Probabilistic Inferences Under Maximum Entropy for Description Logics

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

Marco Wilhelm

Dortmund

2024

Tag der mündlichen Prüfung: 02.09.2024

Dekan: Prof. Dr. Jens Teubner

Gutachterin: Prof. Dr. Gabriele Kern-Isberner

Gutachter: Prof. Dr. Franz Baader

Fakultät für Informatik
Technische Universität Dortmund
Dortmund, Deutschland

"The actual science of logic is conversant at present only with things either certain, impossible, or entirely doubtful, none of which (fortunately) we have to reason on. Therefore the true logic for this world is the calculus of probabilities, which takes account of the magnitude of the probability which is, or ought to be, in a reasonable man's mind."

— James Clerk Maxwell (1850)

Zusammenfassung

In praktischen Anwendungen von wissensbasierten Systemen spielt die Verarbeitung von unsicherem Wissen eine wesentliche Rolle. Die Wahrscheinlichkeitstheorie bietet eine gängige und bewährte Methode, Unsicherheit zu formalisieren und unter ihr verlässliche Schlussfolgerungen zu ziehen. Anwendungen von wahrscheinlichkeitsbasierten Ansätzen der Wissensrepräsentation und -verarbeitung findet man zum Beispiel in der Medizin, unter anderem in der Diagnostik, und den Wirtschaftswissenschaften, beispielsweise bei der Aufdeckung von Betrugsfällen. Ein generelles Problem vieler wahrscheinlichkeitsbasierter Ansätze, wie zum Beispiel von Bayes- und Markov-Netzen, ist jedoch, dass die zugrunde liegende Wahrscheinlichkeitsverteilung vollständig spezifiziert sein muss. Einen Ausweg hierzu bietet das Prinzip der maximalen Entropie (MaxEnt), mit dem es möglich ist, eine unvollständig spezifizierte Wahrscheinlichkeitsverteilung, die üblicherweise in Form einer probabilistischen Wissensbasis gegeben ist, induktiv so aufzufüllen, dass der Informationsgehalt der MaxEnt-Verteilung unter allen möglichen Erweiterungen minimal ist. Dies lässt ein vorsichtiges und prinzipientreues Schlussfolgern zu, wie es in den oben genannten Anwendungsgebieten erforderlich ist.

Für probabilistische Wissensbasen, die aus aussagenlogischen konditionalen Regeln der Form „wenn A gilt, dann folgt B mit einer Wahrscheinlichkeit p“ bestehen, ist die Anwendung des Prinzips der maximalen Entropie bereits recht gut verstanden. In der Praxis sind jedoch meistens unsichere Aussagen von Interesse, die nur über reichhaltigere logische Sprachen zufriedenstellend formuliert werden können. Hierbei handelt es sich insbesondere um Aussagen über Eigenschaften von Individuen und Objekten oder um Aussagen über deren Beziehungen zueinander. Relationale Logiken sowie Beschreibungslogiken sind formale Logiken, die besonders gut geeignet sind, solches Ontologie-Wissen darzustellen. In der vorliegenden Arbeit werden daher wahrscheinlichkeitsbasierte Schlussfolgerungen unter dem Prinzip der maximalen Entropie insbesondere auf Basis von relationalen bzw. beschreibungslogischen probabilisitischen konditionalen Wissensbasen untersucht. Ein zentrales Problem dabei ist, dass mit der Anzahl an betrachteten Individuen und Objekten, den Elementen der Domäne, der zugrunde liegende Wahrscheinlichkeitsraum exponentiell wächst, was das ohnehin als Black-Box-Methode bekannte MaxEnt-Prinzip zu einem wahren „Monstrum“ werden lässt.

Den Kern dieser Arbeit stellt die strukturierte Analyse und die effiziente Behandlung von probabilistischen Wissensbasen für große Domänen unter dem Prinzip der maximalen Entropie dar. Ein wesentlicher Beitrag ist dabei die Entwicklung des „Typed Model Countings“, welches ein formales Rahmenwerk für derartige Analysen bietet. Mit Typed Model Counting können Methoden der Wissenskompilierung, wie sie beispielsweise aus dem „First-Order Model Counting“ bekannt sind, auch auf Konditionale ausgeweitet werden. Ähnlich hochentwickelte Kompilierungsmethoden sind bisher nur sehr punktuell für konditionale Wissensbasen ausgearbeitet worden, zum Beispiel in Form der „Weighted Conditional Impacts“.

Als Ergebnis des Typed Model Countings erhält man eine sehr kompakte Repräsentation derjenigen Information, die benötigt wird, um aus einer Wissensbasis die zugehörige MaxEnt-Verteilung zu berechnen. Diese kompakte Repräsentationsform erfordert neue, Problem angepasste Berechnungsverfahren für die MaxEnt-Verteilung. Daher ist ein zweiter Hauptbeitrag dieser Arbeit das „Condensed Iterative Scaling“, eine Weiterentwicklung des „Generalized Iterative Scalings“, eine numerische Methode zur Annäherung von log-linearen Modellen wie der MaxEnt-Verteilung, die auf die Ausgabe des Typed Model Countings ideal abgestimmt ist.

Neben der Formulierung einer probabilistischen Beschreibungslogik unter der MaxEnt-Semantik, der Logik ALC-ME, auf die hochentwickelte Model-Counting-Strategien angewendet werden um effizient Schlussfolgerungen zu ziehen, werden zudem Schwierigkeiten bei der Betrachtung unendlicher Domänen im Zusammenhang mit dem Prinzip maximaler Entropie angesprochen und es wird ein Lösungsvorschlag gegeben, der auf dem Prinzip „Satisfiability Modulo Theory“ beruht. Insgesamt werden mit dieser Arbeit Methoden bereit gestellt, die es erstmalig ermöglichen, komplexe relationale probabilistische Wissensbasen unter dem Prinzip maximaler Entropie systematisch auszuwerten.

Abstract

In practical applications of knowledge-based systems, handling uncertain knowledge is crucial. Probability theory provides a widely accepted and rigorous framework for formalizing uncertainty and deriving reliable inferences. Probability-based approaches to knowledge representation and reasoning are used in fields such as medicine, for example in diagnostics, and economics, including fraud detection. A common challenge with many probabilistic methods, like Bayesian Networks and Markov Logic Networks, is the need for a fully specified probability distribution. The principle of maximum entropy (MaxEnt) offers a solution to this issue. It enables the inductive completion of an underspecified probability distribution, typically provided as a probabilistic knowledge base, by ensuring that the information content of the MaxEnt distribution is minimal among all possible extensions. This results in cautious and principled reasoning, which is essential in the aforementioned application areas.

The application of the principle of maximum entropy is well-understood for probabilistic knowledge bases consisting of propositional conditional rules of the form "if A holds, then B holds with probability p." However, in practice, one is often interested in uncertain statements which can be formulated satisfactorily by using richer formal languages only. These are, in particular, statements about properties of individuals and objects, as well as their relationships with one another. Relational logics and Description Logics are formal logics well-suited for representing this type of ontological knowledge. Consequently, this thesis explores probability-based reasoning under the principle of maximum entropy for probabilistic conditional knowledge bases formulated over relational logics or Description Logics. A key challenge here is that the underlying probability space grows exponentially with the domain size, i.e., the number of individuals and objects, which transforms the principle of maximum entropy, already known as a black-box approach, into a true "monster" of complexity.

This thesis centers on a structured analysis of and efficient reasoning with probabilistic knowledge bases for large domains under the principle of maximum entropy. A major contribution is the development of "typed model counting", which establishes a formal framework for these analyses. Typed model counting extends knowledge compilation methods, such as those from "first-order model counting", to handle conditional statements effectively. Comparable advanced compilation methods have been developed for conditional knowledge bases only in limited cases, such as in the form of “weighted conditional impacts”.

Typed model counting yields a highly compact representation of the information needed to compute the MaxEnt distribution from a knowledge base. This compact form necessitates new, problem-specific methods for calculating the MaxEnt distribution. Thus, a second major contribution of this thesis is the development of “condensed iterative scaling”, an adaptation of the generalized iterative scaling algorithm, which again is a numerical method designed to approximate log-linear models like the MaxEnt distribution. Condensed iterative scaling is tailored specifically to the output produced by typed model counting.

Eventually, we introduce with ALC-ME a probabilistic Description Logic under a MaxEnt semantics, and apply advanced model counting techniques to enable domain-lifted inferences within ALC-ME. We also address challenges that arise when applying the principle of maximum entropy to infinite domains, proposing a solution based on “satisfiability modulo theory (SMT)”. Overall, this thesis provides methods that, for the first time, enable the structural analysis of complex relational probabilistic knowledge bases under the principle of maximum entropy.

Preface

Writing this thesis has been a rewarding and fulfilling experience, although a challenge that demanded unwavering perseverance. If the reader feels the same, then I consider my efforts successful. Those with a particular interest in Description Logics will need patience, as Description Logics, although they play a central role in this thesis, are primarily addressed in the last part of the thesis. This is because the structure of the thesis reflects the long and complex path my research on maximum entropy reasoning has followed—from propositional background logics to relational logics, and finally to Description Logics. I am confident that the extensive foundational work, especially on relational logics, has proven worthwhile. Not only does it provide valuable insights into lifted inference at maximum entropy, but it also sustains the anticipation for Description Logics throughout.

After thorough research, I can confidently state that if there is one lesson to be learned from uncertain reasoning, it is that nothing is ever truly certain. And so it is by no means certain that a dissertation will be finished once it has begun. The fact that this dissertation is now complete is not solely my accomplishment. I would therefore like to express my heartfelt gratitude to everyone who has supported me and my research over the past years. First and foremost, I am deeply grateful to Prof. Dr. Gabriele Kern-Isberner—not only for her supervision of this thesis but also for inspiring my passion for the field of knowledge representation and reasoning. She granted me the freedom to pursue my research interests in various directions and provided opportunities to present my work at numerous workshops and conferences, where I was fortunate to meet many inspiring people. I am especially thankful for her patience with me and for her unwavering confidence in my work. Finally, I am profoundly appreciative that she welcomed me into her research group, despite my lack of prior contact with her group, her research field, or even with computer science in general.

I would also like to thank Prof. Dr. Franz Baader for his objective and unbiased evaluation of my research as my second supervisor. His insights greatly helped me to enhance my work. It was an honor to collaborate with his group at TU Dresden, particularly with Andreas Ecke, as part of the research project *Hybrid Reasoning for Intelligent Systems* (DFG research group 1513). My heartfelt thanks go to all the wonderful people involved in this project, including those from other universities. I

will always fondly remember our meetings in Aachen, Berlin, Dortmund, Dresden, Freiburg, Leipzig, and Potsdam.

My immediate work environment was both welcoming and supportive, and I am grateful to everyone who was a part of it. First and foremost, I extend my thanks to the staff at Chair 1 at TU Dortmund University and the Knowledge-Based Systems group at FernUniversität in Hagen. I will fondly remember our journal club meetings, Christmas colloquia, mini workshops, and excursions—they were all wonderful experiences that I will surely miss. I would especially like to mention Prof. Dr. Christoph Beierle, an exceptional person and co-author, as well as Diana Howey, Christian Eichhorn, and Alexander Hahn, the best office mates, along with Andre Thevapalan, Meliha Sezgin, Jesse Heyninck, Tanja Bock, Steffen Schieweck, Daan Apeldoorn, Daniela Huvermann, and Patrick Krümpelmann, my irreplaceable colleagues from Dortmund. I am also grateful to Jonas Haldimann, Lars-Phillip Spiegel, Kai Sauerwald, Steven Kutsch, and Marc Finthammer from Hagen, who were always by my side. I would also like to thank Prof. Dr. Falk Howar and Prof. Dr. Emmanuel Müller for their participation in my examination committee.

Last but certainly not least, I would like to thank my family for their support, especially during the final stages of writing this thesis. Nadine and Emilia, I dedicate this work to you!

Marco Wilhelm
Dortmund, 2024

Contents

List of Definitions

List of Propositions

1 Introduction

In the first chapter of this thesis, titled "Probabilistic Inferences Under Maximum Entropy for Description Logics," I outline the scientific context of the topic and provide the motivation for my work. In Section 1.1, I highlight the importance of knowledge-based systems in general and of probabilistic reasoning under the principle of maximum entropy in particular. I also argue why we investigate the principle of maximum entropy in the context of Description Logics here. In Section 1.2, I provide insights into the current state of the art in this research area. From this, I motivate my research questions in Section 1.3, where I also refer to the central results of this thesis. In Section 1.4, I list the preparatory work that led to this thesis, and I describe my own contributions to the listed papers. Section 1.5 presents the structure of the thesis.

1.1 Motivation

Since ChatGPT at the latest, the hype about *artificial intelligence (AI)* has been steadily increasing among the general public [Nathan et al., 2023]. The recent success of AI in practical applications is largely due to the progress made in the field of data-driven AI, in particular in *machine learning* (cf. e.g., [Alpaydin, 2020]). The highly optimized methods that arose in this field are particularly good in recognizing patterns and structures in large amounts of data and are able to make fast and handy predictions based on their findings. However, data-driven AI methods face major challenges, for instance, when it comes to guarantees or explanations of the predictions they have made [Baier et al., 2019]. Thus, principle-based reasoning methods are necessary when reliable statements are required, in particular under uncertain conditions. Examples of critical application areas are diagnosis in medicine [Montgomery, 2018] and fraud detection in economics [Bolton and Hand, 2002].

Knowledge representation and reasoning (KR) is a subfield of artificial intelligence that focuses on representing (uncertain) knowledge in such a way that machines can interpret and use it to make decisions in a principled way (cf., e.g., [Brachman and Levesque, 2004]). The goal of KR is to enable machines to reason about the world in a way similar to humans, by providing them with a structured and formal representation of knowledge. With their internal logical structures and well-motivated inference rules, *knowledge-based systems* (cf., e.g., [Rajendra and Sajja, 2009]) principally have the potential to address the shortcomings of data-driven AI. Nevertheless, knowledge-based systems lead a somewhat shadowy existence. One reason for this may be the lack of efficient computation methods for many knowledge-based approaches. Also, the full potential that arises from the underlying logical structure within a knowledge-based system may not be fully exploited yet.

In this thesis we deal with knowledge-based approaches which encode the uncertainty in knowledge by probabilities. We consider the following scenario from the medical domain as a classical example (cf. the genetics example from [Gelman et al., 2013] as a similar alternative).

Example 1.1.1
A common task of a doctor is to decide, based on a patient's symptoms, which illness the patient is suffering from and what diagnosis the doctor should make. Different combinations of symptoms can often correspond to several clinical pictures, so that a diagnosis can usually be made with some degree of uncertainty only. In order to make this uncertainty tangible, statistics are consulted from which probabilities are derived. However, instead of the probability in which the doctor is interested, namely the probability with which the diagnosis of disease D is correct given the symptoms $S_1, \ldots, S_n$, i.e., the probability $\mathcal{P}(D|S_1 \wedge \ldots \wedge S_n)$, only the probabilities $\mathcal{P}(S_i|D)$ ("What is the probability that the symptom S_i is present in case of disease D?") and $\mathcal{P}(D)$ ("How common is disease D?") are usually available. From a mathematically formalized viewpoint, the doctor's task is now to derive the probability $\mathcal{P}(D|S_1 \wedge \ldots \wedge S_n)$ from his knowledge "$\mathcal{P}(D) = p$ and $\mathcal{P}(S_i|D) = p_i$ for $i = 1, \ldots, n$."

Bayesian Networks [Pearl, 1988] and *Markov Logic Networks* [Richardson and Domingos, 2006] are by far the most prominent approaches to cope with probabilistic inference tasks like the one in Example 1.1.1. Both approaches belong to the graph-based reasoning methods and map conditional dependencies between variables to directed (Bayes) or undirected edges (Markov) of a graph (resp. a network). Such networks compactly represent joint probability distributions over all involved variables by exploiting conditional (in)dependencies. Inference queries can then be answered by computing the relevant probabilities via propagation methods which exploit the internal structure of the network. A disadvantage of these approaches is that the conditional (in)dependencies must be reliably known and hard-coded into the network. Even worse, the probability distributions must be fully specified, usually in form of local distributions. These requirements are not always satisfied in practice, though, like in Example 1.1.1, so that additional assumptions are made. The *Naïve Bayes* approach (cf. e.g., [Lewis, 1998]) is a typical example for that which is often applied to disease prediction problems like the one in Example 1.1.1 (cf. [Jetti et al., 2021; Reddy et al., 2023]).

Example 1.1.2
We illustrate the Naïve Bayes approach by means of Example 1.1.1. In this case, applying Naïve Bayes means to assume that the symptoms $S_1, \ldots, S_n$ are conditionally independent of each other given the disease D, i.e.,

$$\mathcal{P}(S_1 \wedge \ldots \wedge S_n | D) = \prod_{i=1}^{n} \mathcal{P}(S_i | D). \tag{1.1}$$

In general, this is just an approximation, of course, because symptoms usually do not occur independently. With the assumption (1.1) and with *Bayes' theorem*, here stating that

$$\mathcal{P}(D | S_1 \wedge \ldots \wedge S_n) = \frac{\mathcal{P}(D) \cdot \mathcal{P}(S_1 \wedge \ldots \wedge S_n | D)}{\mathcal{P}(S_1 \wedge \ldots \wedge S_n)}$$

holds, the sought probability $\mathcal{P}(D | S_1 \wedge \ldots \wedge S_n)$ is determined by

$$\mathcal{P}(D | S_1 \wedge \ldots \wedge S_n) = \mathcal{P}(D) \cdot \prod_{i=1}^{n} \mathcal{P}(S_i | D)$$

for patients who show the symptoms $S_1, \ldots, S_n$, i.e., for which $\mathcal{P}(S_1 \wedge \ldots \wedge S_n) = 1$ holds. Hence, the doctor is able to estimate the probability of D given $S_1, \ldots, S_n$ based on her knowledge (cf. Example 1.1.1) as required:

$$\mathcal{P}(D | S_1 \wedge \ldots \wedge S_n) = p \cdot \prod_{i=1}^{n} p_i.$$

In this thesis we consider the general case in which the probability distribution from which inferences are to be drawn, i.e., which is supposed to describe the reasoner's *belief state* about the real world, is not fully known. We start from a *knowledge base* consisting of *probabilistic conditionals* $(\psi|\phi)[p]$ representing uncertain statements of the form "if ϕ holds, then ψ follows with probability p." These conditionals constrain the probability distribution but usually do not determine it completely. In principle, inferences could be drawn from all models of the knowledge base together, i.e., from those probability distributions which satisfy the conditionals in the knowledge base. However, drawing inferences in this way is often too uninformative (cf. [Wilhelm et al., 2022]). Instead, we pursue the approach of selecting

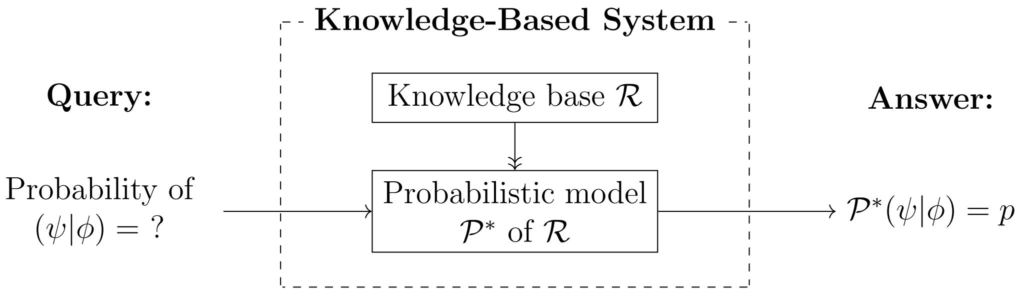

Figure 1.1: Setup of a knowledge-based system for answering probabilistic inference queries in a model-based way.

a distinct probabilistic model of the knowledge base by specifying the ambiguous probability values in a principled way before drawing inferences (cf. Figure 1.1). The selection of a distinct model is known as the *probabilistic model selection task*.

In the propositional context, i.e., when the conditionals $(\psi|\phi)[p]$ are built upon propositional formulas ϕ and ψ, it has been shown that the *principle of maximum entropy (MaxEnt principle)*, which goes back to [Shannon and Weaver, 1949; Jaynes, 1957a,b], provides the only model of a probabilistic conditional knowledge base $\mathcal{R}$ that satisfies some fundamental commonsense principles and, in this sense, solves the model selection task for $\mathcal{R}$ best [Paris, 1998]. The MaxEnt principle states that, when given incomplete probabilistic information about a system, the probability distribution of choice is the one that maximizes the entropy subject to the constraints of the information that is known. Mathematically, this leads to the solution of the nonlinear optimization problem

$$\mathcal{P}^{\mathsf{ME}}_{\mathcal{R}} = \arg\max_{\mathcal{P}\models\mathcal{R}} \ \mathcal{H}(\mathcal{P}), \tag{1.2}$$

where

$$\mathcal{H}(\mathcal{P}) = \sum_{\omega\in\Omega} \mathcal{P}(\omega) \cdot \log \mathcal{P}(\omega)$$

is the *entropy* of $\mathcal{P}$ and the elements in Ω are formalizations of the possible states of the world (aka *possible worlds*). In this way, the maximum entropy model $\mathcal{P}^{\mathsf{ME}}_{\mathcal{R}}$ adds as little information as possible to the knowledge base $\mathcal{R}$ in order to arrive at a complete probability distribution.

Inferences based on the principle of maximum entropy are particularly useful when most cautious yet informative inferences need to be drawn, as it is the case in the applications mentioned above (diagnosis in medicine and fraud detection in economics). While determining the maximum entropy distribution requires solving the non-linear optimization problem (1.2) and, thus, is challenging from a mathematical point of view and commonly known as a black box methodology (cf. Mazzoni [2016]), in the propositional setting the MaxEnt inference problem is theoretically well understood [Jaynes, 1983; Paris and Vencovská, 1990; Paris, 1998, 2006; Kern-Isberner, 2004] and also useful implementations exist (cf., e.g., SPIRIT [Rödder and Meyer, 1996]). In practical applications, however, propositional logics are often not expressive enough to model the knowledge adequately. Usually, one wants to make statements about properties of objects and individuals as well as about relationships between them [Staab and Studer, 2009]. In Example 1.1.1 this could mean the integration of patient-specific data into the background knowledge of the doctor. This inevitably leads to concepts known from *first-order logics* (cf., e.g., [Genesereth and Kao, 2016]) such as constants, which can be used to represent individuals and objects, unary predicates, which are useful to represent properties of individuals and objects, binary predicates, qualified for role relationships, and quantifications. Since first-order logics are known to be undecidable in general and, thus, are usually too powerful to represent and process knowledge, fragments of first-order logics have emerged that are better suited to KR. In their classical form, *Description Logics* [Baader et al., 2008a] represent decidable fragments of first-order logics developed to achieve a good compromise between expressiveness and complexity. Also *relational logics* over finite domains [Genesereth and Kao, 2016], sometimes called *function-free first-order logics*, are widely used in KR.

Thus, the topic of this thesis is to integrate maximum entropy reasoning in relational logics and Description Logics in order to benefit from both the power of principled MaxEnt inferences and the expressiveness of the logics. Therewith, on the one hand, we extend maximum entropy reasoning with the possibility to adequately make statements about individuals, classes of individuals, and relations between them, which is a necessary step towards applying maximum entropy reasoning to real world applications. And, on the other hand, the integration of maximum entropy in relational logics or Description Logics allows us to express uncertain statements within these originally purely classical logics and, therewith, to reason inductively in a nonmonotonic way.

1.2 State of the Art

In the last two decades, a lot of research has been done in the field of *statistical relational AI (StarAI)* [Getoor and Taskar, 2007; De Raedt et al., 2016] on drawing inferences from probability distributions which are defined over relational background languages. In fact, efforts to combine probability theory with (fragments of) first-order logics go back much further. The works of Pearl [Pearl, 1988, 2009] and Halpern [Halpern, 1990, 2005] are among the fundamental ones. Because the probabilities of relational sentences are considered to be the sum of the probabilities of the models of the sentences, especially sophisticated *weighted first-order model counting techniques* (cf. [Van den Broeck et al., 2011; Van den Broeck, 2013; Van den Broeck et al., 2014; Beame et al., 2015]) have contributed to the recent successes in this research area. A common assumption of approaches to StarAI is that the probability distribution is fully specified, hence, the model selection task is usually excluded. Also, the interpretation of probabilistic expressions with free variables, whether formulas or conditionals, is usually left out or based on a simple grounding semantics. This also holds for the probabilistic logic programming language `ProbLog`[1] [De Raedt et al., 2007; De Raedt and Kimmig, 2015] which extends `Prolog` (cf., e.g., [Covington et al., 1988]) by probabilistic facts and, therewith, allows for drawing probabilistic inferences from relational knowledge in the style of declarative programming.

In [Kern-Isberner and Thimm, 2010], propositional conditional maximum entropy reasoning is lifted to the relational setting. This particularly integrates the model selection via (1.2) in the reasoning process. Further, [Kern-Isberner and Thimm, 2010] proposes sophisticated semantics for *open (probabilistic) conditionals* $(\psi|\phi)[p]$, where the relational formulas ϕ and ψ may mention free variables. There is a need to develop semantics for open conditionals because the conditional probability $\mathcal{P}(\psi|\phi)$ is not well-defined in this case. Hence, open conditionals require a richer semantics than interpreting them via conditional probabilities. Open conditionals create real added value, though. While one can argue that relational sentences over finite domains are as expressive as propositional formulas, open conditionals such as $(\mathsf{Flies}(X)|\mathsf{Bird}(X))[p]$ do not have a clear counterpart in propositional probabilistic conditional logics. Instead, the conditional $(\mathsf{Flies}(X)|\mathsf{Bird}(X))[p]$ can be interpreted in many ways. For instance, the conditional can express that "each bird is able to fly with probability p," or "prototypical birds are able to fly with probability p," or "birds in general, e.g., in some kind of mean, are able to fly with probability p." It all depends on the formal semantics of open conditionals.

In this thesis, we rely on the *aggregating semantics* [Kern-Isberner and Thimm, 2010] when interpreting open conditionals. The aggregating semantics combines the

---

[1] `https://dtai.cs.kuleuven.be/problog/` (July 9, 2024)

probabilities of the instances of an open conditional, i.e., the open conditional in which the free variables are substituted by constants, and, therewith, nicely combines statistical and subjective views on the conditional. Basically, a probability distribution $\mathcal{P}$ models a conditional $r = (\psi|\phi)[p]$ under the aggregating semantics if

$$\frac{\sum_{(\psi'|\phi')[p]\in\mathsf{Inst}_\Sigma(r)} \mathcal{P}(\phi' \wedge \psi')}{\sum_{(\psi'|\phi')[p]\in\mathsf{Inst}_\Sigma(r)} \mathcal{P}(\phi')} = p.$$

In essence, probabilities under the aggregating semantics are understood as *degrees of belief* (type 2 probabilities according to [Halpern, 1990]) but also depend on the counts of the instances which are *verified* or *falsified* in a possible world and, thus, also rely on *statistical frequencies* (see [Halpern, 1990; Grove et al., 1994; Bacchus et al., 1996] for different interpretations of probabilities, in particular in first-order settings). An instance $(\psi'|\phi')[p]$ without free variables is *verified* iff both the premise ϕ' and the conclusion ψ' of the conditional are true, and it is *falsified* iff ϕ' is true but ψ' is false.

The aggregating semantics fits well to the principle of maximum entropy because under the aggregating semantics the maximum entropy model of a consistent conditional knowledge base always exists and is unique. As a consequence, I choose the aggregating semantics as the underlying semantics for my investigations on relational maximum entropy. Therewith, my work is in line with the works of Matthias Thimm, Jens Fisseler, and Marc Finthammer [Fisseler, 2010; Thimm et al., 2011; Thimm, 2012; Fisseler, 2012; Finthammer, 2012; Finthammer and Beierle, 2012, 2014; Finthammer, 2017] who developed the ideas from [Kern-Isberner and Thimm, 2010] further. A common theme of these works is to understand the influence of the domain elements on the aggregated MaxEnt probabilities of open conditionals. Technically, this influence can be formalized in the notions of *conditional structures* and of *(weighted) conditional impacts* (cf. [Kern-Isberner, 2004; Finthammer, 2017]) which are ways of evaluating conditionals within possible worlds. In the relational setting, the conditional structure of a possible world is an algebraic representation of how many instances of a conditional are verified or falsified in this world. The verification and falsification behavior can differ both from instance to instance and from possible world to possible world. In [Finthammer, 2017], conditional structures and (weighted) conditional impacts are intensively studied for some simple classes of open conditionals, especially for *atomic conditionals* $(\psi|\phi)[p]$ where ϕ and ψ are atoms. What is still missing, however, is a general framework for the systematic analysis of conditional structures and, thus, of MaxEnt probabilities for more complex conditionals that also contain compounded formulas, in particular under the use of quantifiers.

A further limitation of most approaches to StarAI is that the domain size is considered to be finite in order to ensure that counting techniques work. In Description Logics, the domain size is usually unrestricted, though. The integration of probabilistic reasoning formalisms into Description Logics is not new. Two early extensions of Description Logics by probabilities can be found in [Heinsohn, 1994] and [Jaeger, 1994]. In classical Description Logics, knowledge bases consist of a set of terminological axioms describing relationships between concepts, and a set of assertional axioms stating concept and role assertions of (tuples of) individuals. Both approaches to probabilistic Description Logics, [Heinsohn, 1994] and [Jaeger, 1994], allow for probabilistic terminological axioms of the form $\mathcal{P}(D|C) = p$ in addition, where the semantics is based on probability measures on the set of all concept descriptions (modulo equivalence). Therewith, the semantics of probabilistic terminological axioms has a statistical origin. Further, both approaches [Heinsohn, 1994] and [Jaeger, 1994] do not involve a model selection strategy but draw inferences from all models of the probabilistic Description Logic knowledge base. In addition, [Jaeger, 1994] allows for probabilistic assertions of the form $\mathcal{P}(C(a)) = p$ which express a degree of belief in a being an instance of C. In order to connect the statistical probability measures over concept descriptions and the subjective degrees of belief in assertions, [Jaeger, 1994] uses *cross-entropy minimization* to find measures for probabilistic assertional axioms which do not violate the assertions and are closest to the measures for the probabilistic terminological axioms.

In [Koller et al., 1997] probabilistic terminological knowledge is considered where the probability distributions are defined over the properties of individuals expressing to which extent these properties overlap. The probability distributions are represented as Bayesian Networks. This allows for tractable reasoning about classes of individuals but there is no possibility to include assertional axioms.

The probabilistic Description Logic considered in [Gutiérrez-Basulto et al., 2011] includes probabilistic concept and role constructors. A probabilistic concept expresses a degree of belief in individuals being an instance of the concept and probabilistic roles are defined analogously. Probabilities are also applied to compounded concepts but in this logic it is not possible to express probabilistic terminological knowledge like probabilistic concept inclusions. In [Gutiérrez-Basulto et al., 2017] the probabilistic Description Logic is extended by probabilistic conditionals $\mathcal{P}_{\sim n}(C|D)$ with $\sim \in \{\leq, <, =, >, \geq\}$ which, however, have a different semantics to conditionals in this thesis. Probabilistic conditionals $\mathcal{P}_{\sim n}(C|D)$ from [Gutiérrez-Basulto et al., 2017] are concept constructors and represent all domain elements d for which the degree of belief $p_d^{\mathcal{I}}(C \sqcap D)$ in d being an instance of both C and D, and the degree of belief $p_d^{\mathcal{I}}(D)$ in d being an instance of D satisfy the relation $p_d^{\mathcal{I}}(C \sqcap D) \sim n \cdot p_d^{\mathcal{I}}(D)$.

Knowledge bases in [Niepert et al., 2011] consist of classical (deterministic) terminological knowledge as well as weighted knowledge where the weights w_C are real numbers assigned to terminological axioms C. The probabilistic interpretation of such weighted axioms is via log-linear probability distributions where the weights of the axioms are considered as the weights of the log-linear models. More precisely, let $(\mathcal{C}^D, \mathcal{C}^U)$ be a knowledge base in the sense of [Niepert et al., 2011], i.e., $\mathcal{C}^D$ is the deterministic knowledge and $\mathcal{C}^U$ the weighted knowledge, then the probability of a weighted axiom C' is given by

$$\mathcal{P}(C') = \frac{1}{Z} \exp\left(\sum\nolimits_{(C,w_C)\in\mathcal{C}^U :\, C' \models C} w_C\right) \tag{1.3}$$

provided that C' is coherent and consistent with $\mathcal{C}^D$, and by $\mathcal{P}(C') = 0$ otherwise, where Z is the normalization constant of $\mathcal{P}$. Since the maximum entropy model is also a log-linear model, the approach from [Niepert et al., 2011] is the approach to probabilistic Description Logics which is most similar to ours. The main difference is how the weights go into the log-linear model since [Niepert et al., 2011] does not rely on the aggregating semantics (compare (1.3) to (4.13) where the exp-term mentions so-called *feature functions* which combine conditional structures and the probabilities of conditionals as an outcome of the aggregating semantics). In addition, [Niepert et al., 2011] does not allow for assertional knowledge.

Another probabilistic Description Logic, $\mathcal{ALCP}$, is presented in [Peñaloza and Potyka, 2016]. $\mathcal{ALCP}$ makes explicit use of the principle of maximum entropy. However, in $\mathcal{ALCP}$ probabilities are not assigned to Description Logic axioms directly but to an additional propositional logic which serves as a context for the Description Logic. Therewith, the language $\mathcal{ALCP}$ is quite different to our approach. In $\mathcal{ALCP}$ no assertional axioms are allowed either. Further work on probabilistic Description Logics can be found, for example, in [Lukasiewicz, 2008; Sebastiani, 1994; Klinov, 2011; Klinov and Parsia, 2013; Riguzzi et al., 2013], and also the survey papers [Lukasiewicz and Straccia, 2008; de Salvo Braz et al., 2008] provide additional information about probabilistic Description Logics and probabilistic first-order logics in general. However, none of these approaches rely on the aggregating semantics or make explicit use of the principle of maximum entropy in the way we do.

In our approach on systematically analyzing relational maximum entropy reasoning, we make use of model counting techniques similar to those from StarAI. When transferring our results to Description Logics, we stick to a fixed finite domain in order that the model counting techniques still work in this case. To maintain the possibility of expressing statements over infinite domains, we draw on an approach called *satisfiability modulo theory* [Barrett et al., 2021]. It generalizes the satisfiability problem (SAT) to more complex formulas, in our case involving integers and real numbers. The satisfiability of these expressions is tested with respect to an arithmetic background theory. Our approach is closely linked to Description Logics with concrete domains [Baader and Hanschke, 1991; Baader and Bortoli, 2023, 2024].

1.3 Research Questions and Main Contributions

The application of maximum entropy reasoning to Description Logics with the goal to draw principled probabilistic inferences from relational domain knowledge raises many questions and research tasks. In this thesis, we mainly tackle the following four research questions:

Q1 Is there a systematic and efficient way of computing conditional structures with respect to relational probabilistic conditional knowledge bases even for huge domain sizes?

Background: Under the *aggregating semantics* the *maximum entropy model* of a relational probabilistic conditional knowledge base $\mathcal{R}$ provides a product representation which is fully determined by the *conditional structures* with respect to the conditionals in $\mathcal{R}$ and their probabilities (cf. [Kern-Isberner and Thimm, 2010; Jaynes, 1983]). Hence, computing conditional structures is essential for maximum entropy reasoning. In [Finthammer, 2017], it has been shown that an efficient computation of conditional structures is a tough task, especially if the domain size k is huge, because the number of possible worlds for which the conditional structures have to be computed grows exponentially in k, and also the conditional structures themselves become more complex.

Q2 Is there a systematic and efficient way of drawing inferences at maximum entropy from relational probabilistic conditional knowledge bases even for huge domain sizes?

Background: The question that directly follows from **Q1** is, of course, whether the maximum entropy model of a knowledge base $\mathcal{R}$ can also be computed and used to draw inferences efficiently. The maximum entropy model can be stored in form of a real-valued vector, which we call the ME-*vector*, the length of which is constant in the domain size (cf. [Kern-Isberner and Thimm, 2010; Finthammer, 2017]). It is unclear under which conditions this representation can be computed and used to draw inferences at maximum entropy while preventing an exponential blowup caused by the domain size.

Q3 How to combine maximum entropy reasoning and Description Logics with fixed finite domains?

Background: Classical *Description Logics* (cf. [Baader et al., 2007]) are fragments of first-order logics and, therefore, related to the relational background logics of conditionals that are addressed in [Kern-Isberner and Thimm, 2010] as well as in **Q1** and **Q2**. At first sight, the investigations on relational maximum entropy reasoning should be transferable to Description Logics. However, the approach in [Kern-Isberner and Thimm, 2010] builds upon an *Herbrand* *semantics* and is limited to finite domains which is different from common Description Logics. A reasonable first step towards integrating maximum entropy reasoning into Description Logics is to investigate Description Logics with fixed finite domains. Besides probabilities, this particularly introduces conditionals to Description Logics and an important question is how the aggregating semantics of open probabilistic conditionals transfers to conditionals defined over Description Logics.

Q4 How to deal with infinite domains in maximum entropy reasoning?

Background: Description Logics allow for countably infinite domains, in principle. While considering fixed finite domains as in **Q3** is a common simplification in probabilistic Description Logics [Gaggl et al., 2016], we also want to investigate the case of infinite domains here. This is a challenging task when relying on the principle of maximum entropy because the maximum entropy model tends to the uniform distribution by fulfilling the paradigm of *conditional indifference* [Kern-Isberner and Thimm, 2012]. As a consequence, in the absence of beliefs the probability distribution with maximal entropy should be the uniform distribution. It is well known that there is no uniform distribution in countably infinite probability spaces, though.

The main contributions of this thesis to answer the research questions **Q1** to **Q4** are the following.

Typed Model Counting: By far the most important contribution of this thesis is the development of *typed model counting* in Section 6.2 and the subsequent sections. With typed model counting it is possible to exploit (algebraic) first-order model counting strategies [Kimmig et al., 2017; Van den Broeck, 2013] when computing conditional structures or related expressions. Therewith, we fruitfully link conditional reasoning based on maximum entropy to the research field of *first-order model counting* and *knowledge compilation* [Darwiche and Marquis, 2002] and make use of the sophisticated results and strategies from this line of research. A challenging task when computing conditional structures for relational knowledge bases over huge domains is the application of combinatorial arguments with respect to indistinguishabilities between the domain elements, which has been limited to some special cases until now (cf. [Finthammer, 2017]). Typed model counting, instead, allows to incorporate combinatorial arguments for complex relational knowledge bases which, for example, mention quantifiers. Typed model counting can also be used to draw inferences at maximum entropy and, thus, we address the research questions **Q1** and **Q2** with this approach.

Condensed Iterative Scaling: The output of typed model counting when used to compute conditional structures is a very compact representation which aggregates conditional structures of all conditionals from the knowledge base over all possible worlds. In order to benefit from this representation, it is necessary to develop new, problem-adapted numerical methods for computing the respective maximum entropy model thereof. With *condensed iterative scaling* we provide such a problem-adapted algorithm. We present and discuss condensed iterative scaling in Section 5.2. Like typed model counting, this contribution is mainly related to the research questions **Q1** and **Q2**.

Description Logic $\mathcal{ALC}^{\mathsf{ME}}$: In Section 7.1, we propose the probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ which integrates probabilistic conditionals into the prototypical Description Logic $\mathcal{ALC}$. Conditionals in $\mathcal{ALC}^{\mathsf{ME}}$ are interpreted based on the aggregating semantics and the principle of maximum entropy. Hereby, we stick to fixed finite domains and an Herbrand-based semantics. Therewith, we combine maximum entropy reasoning and Description Logics as promised in **Q3**. We discuss how $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases can be translated into relational probabilistic conditional knowledge bases in order to benefit from the concepts of typed model counting and condensed iterative scaling, and we also show how to draw domain-lifted inference in $\mathcal{ALC}^{\mathsf{ME}}$ (Section 7.2).

SMT-Approach for Infinite Domains: In Section 7.3, we address the research question **Q4** by introducing a novel concept constructor into $\mathcal{ALC}^{\mathsf{ME}}$ which is based on linear arithmetic constraints and closely linked to the concept of concrete domains in Description Logics [Baader and Bortoli, 2023, 2024]. The linear arithmetic constraints are formulated over the integers or real numbers and, therewith, allow for statements about a countably infinite domain (in case of integers) and also an uncountably infinite domain (in case of reals). The satisfaction of concepts that involve linear arithmetic constraints is decided modulo the theory of linear arithmetic. As a consequence, the probability space remains finite and the principle of maximum entropy can be applied.

1.4 Previous Publications

An overview of the peer-reviewed publications to which I have contributed, and which served as preparatory work for this thesis, follows. For each publication, I note what parts of this publication can be attributed to me. Further publications by me with less reference to this thesis are grouped by their topic and briefly discussed afterwards. The citations of the form [pX] refer to Section A.3 which provides a complete list of my peer-reviewed publications until the day of submission of the thesis.

Systematic Computation of Conditional Structures: In the paper [p3], we developed an algorithm called CondStructor which computes the conditional structures of possible worlds with respect to propositional conditional knowledge bases. Conditional structures are automatically grouped to equivalence classes. The basic concept used in this paper is a variant of *Boole's expansion theorem* (also known as *Shannon decomposition*; [Boole, 1854; Shannon, 1949]). The system can be seen as a precursor to (propositional) *typed model counting.*

I am the main author of this paper, the essential ideas of CondStructor are mine. Co-authors of this paper are Gabriele Kern-Isberner and Andreas Ecke.

Propositional Typed Model Counting: In the paper [p5], we presented the concept of *typed model counting* for probabilistic conditional reasoning at maximum entropy for the first time. Central in this paper is the compilation of conditional knowledge bases into structured sentences in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$ normal form which can then be used to compute conditional structures by counting their typed models. In this paper, the approach of *typed model counting* is applied to a propositional setting.

I am the main author of this paper. The idea of *typed model counting* and its realization by structured sentences are my work. Co-author of this paper is Gabriele Kern-Isberner.

Relational Maximum Entropy and Basic Independence Results: In the paper [p8], we have proven some basic independence results for relational maximum entropy reasoning. The main outcomes of this paper are product representations of the maximum entropy distribution which exploit splittings in the syntax of the underlying knowledge base on the one hand and isomorphic operands, basically caused by indistinguishabilities between domain elements, on the other hand.

I am the main author of this paper. I have contributed the formulations and the proofs of the central propositions in the paper. Co-authors of this paper are Gabriele Kern-Isberner and Andreas Ecke.

First-Order Typed Model Counting: In the paper [p7], we lifted the concept of *typed model counting* (cf. [p5]) to the relational setting. It constitutes one of the essential sources for this thesis.

I am the main author of this paper. The idea and the technical elaboration of *first-order typed model counting* are mine. Co-authors of this paper are Marc Finthammer, Gabriele Kern-Isberner, and Christoph Beierle.

Generalized Iterative Scaling: In the paper [p10], we proposed an iterative scaling algorithm called iGIS for calculating the maximum entropy distribution of relational probabilistic conditional knowledge bases. Our algorithm continues work from [Finthammer, 2012; Finthammer and Beierle, 2012] by exploiting independence results for the maximum entropy distribution (cf. also [p8]). Methodically, we extended *weighted conditional impacts (WCIs)* to *WCI systems* by respecting probabilistic independencies.

I am the main author of this paper. The concept of *weighted conditional impacts* and the idea of using them for generalized versions of iterative scaling algorithms for maximum entropy computations can be attributed to Marc Finthammer. My contribution is lifting WCIs to WCI systems and adapting the iterative scaling algorithm accordingly. The implementation of iGIS is my work. Co-authors of this paper are Gabriele Kern-Isberner, Marc Finthammer, and Christoph Beierle.

Condensed Iterative Scaling: The contributions of [p13] are manifold. Besides the further elaboration of *first-order typed model counting* (cf. [p7]), we presented with *condensed iterative scaling* an algorithm which merges the research lines of improving iterative scaling algorithms for maximum entropy reasoning and of *typed model counting*. In addition, we established a connection between maximum entropy reasoning and *Markov Logic Networks* [Richardson and Domingos, 2006]. This paper constitutes another important source of this thesis.

I am the main author of this paper. The *condensed iterative scaling algorithm* is my work. The idea of combing maximum entropy reasoning and *Markov Logic Networks* is from Gabriele Kern-Isberner. Co-authors of this paper are Gabriele Kern-Isberner, Marc Finthammer, and Christoph Beierle.

Description Logic $\mathcal{ALC}^{\mathsf{ME}}$: In the paper [p12], we presented the probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ which introduces probabilistic conditionals into the prototypical Description Logic $\mathcal{ALC}$. We investigated $\mathcal{ALC}^{\mathsf{ME}}$ knowledge bases without assertions with respect to a fixed finite domain under a maximum entropy semantics. We were able to show that drawing inferences is domain-liftable in this case.

I am the main author of this paper. The foundations on Description Logics were provided by Franz Baader and Andreas Ecke. I myself have drawn the connection to maximum entropy reasoning and provided the proofs of the paper. Co-authors of this paper are Gabriele Kern-Isberner, Andreas Ecke, and Franz Baader.

Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ and Typed Model Counting: In the paper [p14], we applied typed model counting to the probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$. We discussed several illustrating examples, among others the *antecedent conjunction problem* (cf. also [Kern-Isberner, 2001a]).

I am the main author of this paper. In particular, I contributed the calculations of lower and upper bounds for the maximum entropy vector with respect to the *antecedent conjunction problem.* Co-author of this paper is Gabriele Kern-Isberner.

Complexity Results for $\mathcal{ALC}^{\mathsf{ME}}$: In the paper [p15], we investigated the complexity of deciding the consistency of knowledge bases in the probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$. Among other results, we proved that the domain size complexity of deciding the consistency of an $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base is in P for unary encodings of the domain size.

The main author of this paper is Franz Baader. Besides Andreas Ecke and Gabriele Kern-Isberner, I co-authored with minor contributions.

Maximum Entropy and Linear Arithmetic Constraints: In the paper [p27], we tackled the problem of combining maximum entropy reasoning and infinite domains. Because the maximum entropy distribution is not well-defined for countably infinite domains, we proposed an approach based on satisfiability modulo theory (SMT) which outsources the analysis of infinite structures to an underlying linear arithmetic theory. In Section 7.3 we lift this approach from propositional conditionals modulo linear arithmetic to conditionals with description-logical expressions.

I am the only author of the paper. All contributions of this paper are my work.

Below are the publications to which I have contributed but which have only a minor connection to the thesis.

Maximum Entropy and Gröbner-Basis Theory: In the papers [p1], [p2], [p4], and [p9], we investigated the maximum entropy approach on a more abstract, purely symbolic level and asked for inferences that can be drawn from a probabilistic conditional knowledge base at maximum entropy without having to solve the maximum entropy optimization problem numerically. We showed that, in some cases, maximum entropy inferences can be drawn symbolically by using *Gröbner basis theory* (cf. [Becker and Weispfenning, 1993]) from the field of symbolic computation.

Conditional Reasoning and ACT-R: The papers [p18], [p20], [p21], [p22], [p24], and [p29] contribute to the innovative research field of *cognitive logics* [Ragni et al., 2020] that deals with the connection of formal approaches to uncertain reasoning and cognitive science. In the mentioned papers, we combined conditional reasoning and the cognitive architecture *ACT-R* (*Adaptive Control of Thought-Rational*, cf. [Anderson et al., 2004]). While most of the works concentrate on qualitative reasoning, a connection between ACT-R and the principle of maximum entropy is worked out in [p22].

Conditionals with Default Negation: The papers [p6] and [p16] introduce the concept of *default negation* [Clark, 1977] to conditionals. The default negation “not A” is a form of weak negation and is intended to express that “not A” holds by default as long as A is not known to be true. While [p6] investigates qualitative conditionals with default negation, [p16] deals with probabilistic conditionals and default negation under the principle of maximum entropy.

Maximum Entropy and Belief Fusion: The paper [p17] applies the maximum entropy principle for relational conditionals to the task of fusing beliefs of multiple agents. Here, closed conditionals which mention a single constant represent the beliefs of an individual agent and open conditionals without constants stand for the fused beliefs.

Maximum Entropy and Syllogism Tasks: In [p19], the principle of maximum entropy is used to predict human responses to *syllogism tasks* (cf. [Khemlani and Johnson-Laird, 2012]). Syllogisms are specific types of semi-logical conclusions. An example is: “All cats have fur. Some pets are cats. Hence, some pets have fur.” In syllogism tasks, humans are asked to choose the most appropriate conclusion from given preconditions out of a series of prescribed options.

Qualitative Uncertain Reasoning: The papers [p11], [p23], [p25], [p26], [p28], [p30], [p31], and [p32] are dedicated to qualitative uncertain reasoning with different objectives.

I was thankfully allowed to incorporate my scientific work on probabilistic inferences under maximum entropy into a tutorial and, thus, share it with a broader audience as well. Together with Tanya Braun and Marcel Gehrke, I gave a tutorial on the topic of “Statistical Relational AI – Exploiting Symmetries” at the KR 2023 conference.[2]

---

[2] Tutorial website: `https://www.uni-muenster.de/Informatik.AGBraun/en/research/tutorials/kr-23.html` (July 9, 2024)

1.5 Outline

The remainder of this thesis is organized as follows.

Chapter 2: In Chapter 2, we settle the logical foundations of our investigations on probabilistic inferences under maximum entropy for Description Logics. After a general introduction to propositional logics (Section 2.1) and relational logics (i.e., function-free first-order logics over finite domains; Section 2.2), we extend relational logics by domain and set constraints in Section 2.3. Basically, domain and set constraints allow us to subtly restrict the scope of quantifications and open conditionals. In addition, domain and set constraints are useful for model counting. In Section 2.4, an introduction to Description Logics follows, in particular to the Description Logic $\mathcal{ALC}$, before we conclude this chapter with a comparison of the logics in Section 2.5.

Chapter 3: Chapter 3 deals with the representation of (uncertain) knowledge in form of relational conditionals. In Section 3.1, we introduce basic notions of relational conditionals and deepen the discussion of conditional structures in Section 3.2. Conditional structures play a central role in the systematic computation of maximum entropy models of probabilistic knowledge bases. We extend relational conditionals by probabilities in Section 3.3. There, we also study the semantics of so-called closed conditionals which are relational probabilistic conditionals built upon formulas without free variables. While the semantics of closed conditionals is straightforward, the semantics of open conditionals with free variables is ambiguous. In Section 3.4, we discuss different semantics of open conditionals with a focus on the aggregating semantics on which we rely in this thesis.

Chapter 4: In Chapter 4, we discuss the principle of maximum entropy for relational probabilistic conditional knowledge bases. This principle allows us to infer unique and principled probabilistic models of knowledge bases in Section 4.2. Beforehand, we formally define knowledge bases and the in this thesis central probabilistic inductive inference task (Section 4.1) from which we derive the maximum entropy inference task. After that, in Section 4.3, we discuss several formulations of the maximum entropy optimization problem. What is worth mentioning here is the product representation of the maximum entropy distribution which results from the dual maximum entropy optimization problem and which revisits the concept of conditional structures. The possibility to express the maximum entropy distribution in form of a product is the fundamental prerequisite for our further investigations on the systematic and efficient computation of maximum entropy probabilities. Section 4.4 is an excursion on maximum entropy and Markov Logic Networks.

Chapter 5: In Chapter 5, we define the problem of lifted inference at maximum entropy. Lifted inference deals with the question how to draw inferences from knowledge about large domains efficiently. In the narrower complexity theoretical sense, lifted inference means answering inference queries in time at most polynomial in the number of domain elements. Here, we want to understand the lifted inference task more generally as the systematic analysis of the influence of the domain size on the inference and the use of these findings to efficiently answer inference queries. We aim at structural effects, in particular symmetries caused by indistinguishable domain elements, which provide compact representations of the information that is necessary to calculate maximum entropy inferences. Therefore, in Section 5.1, we conceptualize the term lifted inference. Afterwards, in Section 5.2, we deal with the numerical calculation of the maximum entropy model and propose an algorithm called condensed iterative scaling that specifically aims to encapsulate the domain size as well as possible in order to make it manageable.

Chapter 6: Chapter 6 is the central chapter of this thesis and introduces the concept of typed model counting. Typed model counting is a modification of model counting, here specifically first-order model counting, which aims to compactly represent the conditional structure of a knowledge base, and thereby provides the input for the condensed iterative scaling algorithm. Typed model counting raises the model counting of first-order sentences to the level of conditionals. In Section 6.1, we recall the basic ideas of first-order model counting and, in particular, the normal form sd-DNNF. Sentences in sd-DNNF normal form are particularly well-suited to count their models. In Section 6.2, we then discuss the concept of typed model counting in detail and show its connection to algebraic model counting in Section 6.3. Eventually, we apply typed model counting to the task of computing the conditional structures of knowledge bases in Section 6.4.

Chapter 7: In Chapter 7, we transfer our results from Chapter 6 to probabilistic conditional knowledge bases that are built upon description-logical expressions. We define the Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ and introduce maximum entropy reasoning for $\mathcal{ALC}^{\mathsf{ME}}$ in Section 7.1. In Section 7.2, we investigate lifted inferences at maximum entropy from $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases. We also deal with the question of what happens when the size of the domain is infinite (Section 7.3). We point out difficulties that arise in the interaction between the principle of maximum entropy and infinite domains and propose an approach on how to integrate statements about infinite domains into $\mathcal{ALC}^{\mathsf{ME}}$ by exploiting the concept of satisfiability modulo theory (SMT).

Chapter 8: In Chapter 8, we draw a conclusion to this thesis. We briefly summarize the content of this thesis in Section 8.1, and discuss the main results while pointing to future work in Section 8.2.

Appendix: The appendix contains a separate section for the central mathematical foundations of this thesis (Section A.1), a catalog of knowledge bases that have been examined using the method of typed model counting (Section A.2), as well as a list of peer-reviewed publications to which I contributed as an author or co-author (Section A.3). The thesis ends with the mandatory bibliography.

2 Logical Foundations

In this chapter, we settle the logical foundations of this thesis. In Section 2.1, we begin with a brief introduction to *propositional logics* (cf., e.g., [Hurley and Watson, 2017]) which constitute a common basis for many logical systems in *knowledge representation and reasoning (KR)*. After that, we discuss *relational logics* (cf. e.g., [Genesereth and Kao, 2016; Lewis, 1918; Ullman, 1988]) which extend propositional logics mainly by the concepts of constants and predicates and, therewith, allow for a more adequate representation of knowledge about individuals, their properties, and relations among them than propositional logics can do (Section 2.2). We extend relational logics by quantifiers with domain and set constraints in Section 2.3. In Section 2.4 we give an overview of *Description Logics* [Baader et al., 2007, 2017] which, like relational logics, constitute fragments of *first-order logics*. Thereby, we focus on the prototypical Description Logic $\mathcal{ALC}$. Finally, we compare the different logics, particularly with regard to their suitability to represent knowledge, in Section 2.5.

2.1 Propositional Logics

Propositional logics (cf., e.g., [Hurley and Watson, 2017]) are formal logics that allow one to reason about the truth values of statements which can either be true or false. They provide a formal language in which such statements can be expressed, as well as an inference formalism with which statements can be deduced from others. Propositional logics serve as the basis for many more elaborated logics and reasoning formalisms.

In this section, we formally define the basic concepts of *propositional logics* and fix related notations, abbreviations, and conventions that will be used throughout the thesis. Our deliberations loosely follow the introduction to propositional logics in [Beierle and Kern-Isberner, 2014].

▶ Syntax and Semantics of Propositions

We consider a *propositional language* $\mathcal{L}(\Sigma_P)$ which is defined over a finite set of *atomic propositions* Σ_P. Atomic propositions serve as the basic building blocks of $\mathcal{L}(\Sigma_P)$. The set Σ_P is also called *propositional signature*. Statements, that can either be true or false, are represented in $\mathcal{L}(\Sigma_P)$ as *propositional formulas* which are the propositional atoms in Σ_P or compounded propositional formulas obtained by the use of the common connectives *negation* ($\neg$), *conjunction* ($\wedge$), or *disjunction* ($\vee$). Later, we also introduce the *material implication* ($\Rightarrow$) and *material equivalence* ($\Leftrightarrow$) as additional connectives.

Definition 2.1.1: Propositional Language
(cf. e.g., [Beierle and Kern-Isberner, 2014])

Let Σ_P be a finite set called *propositional signature*. The *propositional language* $\mathcal{L}(\Sigma_P)$ is the set of *propositional formulas* over Σ_P, or *propositions* for short, which are either *atomic propositions* from Σ_P or compounded propositional formulas of the form $\neg\phi$ (*negation*), $(\phi \wedge \psi)$ (*conjunction*), or $(\phi \vee \psi)$ (*disjunction*) where ϕ and ψ are propositional formulas from $\mathcal{L}(\Sigma_P)$.

The semantics of propositional formulas in $\mathcal{L}(\Sigma_P)$ is given by *propositional interpretations* which assign to each proposition a truth value from $\{0, 1\}$. The truth value 1 stands for *true*, the truth value 0 for *false*.

Definition 2.1.2: Propositional Interpretation
(cf. e.g., [Beierle and Kern-Isberner, 2014])

A *propositional interpretation* is a mapping $I_P\colon \mathcal{L}(\Sigma_P) \to \{0, 1\}$ which assigns to every propositional formula in $\mathcal{L}(\Sigma_P)$ a truth value, either 1 (*true*) or 0 (*false*). Propositional interpretations are determined by their assignments to the atomic propositions in Σ_P. With $\phi, \psi \in \mathcal{L}(\Sigma_P)$, the interpretation of compounded propositional formulas is recursively given by

- $I_P(\neg\phi) = \begin{cases} 1 & \text{if} \quad I_P(\phi) = 0 \\ 0 & \text{otherwise} \end{cases}$,
- $I_P((\phi \wedge \psi)) = \begin{cases} 1 & \text{if} \quad I_P(\phi) = 1 \quad \text{and} \quad I_P(\psi) = 1 \\ 0 & \text{otherwise} \end{cases}$,

I_i	$I_i(S)$	$I_i(R)$	$I_i(B)$	$I_i(\phi_{\mathsf{bow}})$	I_i	$I_i(S)$	$I_i(R)$	$I_i(B)$	$I_i(\phi_{\mathsf{bow}})$
I_1	1	1	1	1	I_5	0	1	1	0
I_2	1	1	0	1	I_6	0	1	0	1
I_3	1	0	1	0	I_7	0	0	1	0
I_4	1	0	0	1	I_8	0	0	0	1

Table 2.1: Evaluation of the proposition $\phi_{\mathsf{bow}} = ((S \wedge R) \vee \neg B)$ from Example 2.1.3.

- $I_P((\phi \vee \psi)) = \begin{cases} 1 & \text{if} \quad I_P(\phi) = 1 \quad \text{or} \quad I_P(\psi) = 1 \\ 0 & \text{otherwise} \end{cases}.$

Because propositional interpretations are determined by their assignments to the atomic propositions, there are $2^{|\Sigma_P|}$-many different propositional interpretations over Σ_P. We denote the set of all propositional interpretations over Σ_P with $\mathsf{Int}(\Sigma_P)$.

Example 2.1.3

We consider the propositional signature $\Sigma_P = \{\mathsf{Sunshine}, \mathsf{Rain}, \mathsf{Rainbow}\}$ and the propositional formula $\phi \in \mathcal{L}(\Sigma_P)$ given by

$$\phi_{\mathsf{bow}} = ((\mathsf{Rain} \wedge \mathsf{Sunshine}) \vee \neg \mathsf{Rainbow}).$$

> "It rains and the sun shines, or there is no rainbow."

The interpretations in $\mathsf{Int}_P(\Sigma_P)$ and their evaluation of ϕ_{bow} is shown in Table 2.1 where we use the abbreviations $S = \mathsf{Sunshine}$, $R = \mathsf{Rain}$, and $B = \mathsf{Rainbow}$. For instance, ϕ_{bow} is true in the interpretation I_1 which maps all three atomic propositions Rain, Sunshine, and Rainbow to true, because the conjunction $(\mathsf{Rain} \wedge \mathsf{Sunshine})$ which occurs in ϕ_{bow} is true in I_1. Therewith, the disjunction in ϕ_{bow} is true as well, independently of the evaluation of its second disjunct $\neg\mathsf{Rainbow}$.

A propositional interpretation $I_P \in \mathsf{Int}(\Sigma_P)$ which maps a formula $\phi \in \mathcal{L}(\Sigma_P)$ to true, $I_P(\phi) = 1$, is called a *propositional model* of ϕ. The set of all propositional models of ϕ is denoted with $\mathsf{Mod}_{\Sigma_P}(\phi)$. If a propositional formula $\phi \in \mathcal{L}(\Sigma_P)$ *entails* a formula $\psi \in \mathcal{L}(\Sigma_P)$, i.e., if $\mathsf{Mod}_{\Sigma_P}(\phi) \subseteq \mathsf{Mod}_{\Sigma_P}(\psi)$ holds, then we write $\phi \models_{\Sigma_P} \psi$. Logical entailment is the inference formalism in propositional logics. If ϕ and ψ entail each other, $\mathsf{Mod}_{\Sigma_P}(\phi) = \mathsf{Mod}_{\Sigma_P}(\psi)$, then ϕ and ψ are *logically equivalent*, in symbols $\phi \equiv_{\Sigma_P} \psi$.

Example 2.1.4
We consider the signature $\Sigma_P = \{\mathsf{Sunshine}, \mathsf{Rain}, \mathsf{Rainbow}\}$ and the formula

$$\phi_{\mathsf{bow}} = ((\mathsf{Rain} \wedge \mathsf{Sunshine}) \vee \neg\mathsf{Rainbow})$$

from Example 2.1.3 again. The set of the propositional models of ϕ_{bow} is (cf. Table 2.1)

$$\mathsf{Mod}_{\Sigma_P}(\phi_{\mathsf{bow}}) = \{I_1, I_2, I_4, I_6, I_8\}.$$

The subformula $\psi_{\mathsf{bow}} = (\mathsf{Rain} \wedge \mathsf{Sunshine})$ of ϕ_{bow} entails ϕ_{bow} because

$$\mathsf{Mod}_{\Sigma_P}(\psi_{\mathsf{bow}}) = \{I_1, I_2\} \subseteq \mathsf{Mod}_{\Sigma_P}(\phi_{\mathsf{bow}}).$$

It is easy to prove via truth tables such as Table 2.1 that ϕ_{bow} and ϕ'_{bow}, where

$$\phi'_{\mathsf{bow}} = ((\mathsf{Rain} \vee \neg\mathsf{Rainbow}) \wedge (\mathsf{Sunshine} \vee \neg\mathsf{Rainbow})),$$

are logically equivalent.

We assume that the reader is familiar with the common laws that preserve logical equivalence when rearranging propositional formulas, like the *distributive law*

$$((\phi \wedge \phi') \vee \psi) \equiv_{\Sigma} ((\phi \vee \psi) \wedge (\phi' \vee \psi))$$

for $\phi, \phi', \psi \in \mathcal{L}(\Sigma_P)$, for instance. Actually, by applying the distributive law to ϕ_{bow} from Example 2.1.3, one can show that the formulas ϕ_{bow} and ϕ'_{bow} (cf. Example 2.1.4) are logically equivalent as well.

▶ Abbreviations and Conventions

Throughout this thesis, we use the following abbreviations and conventions in order to improve the readability of logical expressions.

- If it is clear from the context which signature Σ_P is used, then we omit the subscript Σ_P. For example, we write $\models$ instead of $\models_{\Sigma_P}$ for the entailment relation, and $\equiv$ instead of $\equiv_{\Sigma_P}$ to indicate logical equivalence.
- We stick to the common conventions when omitting parentheses in propositional formulas, always provided that the formulas are represented unambiguously. For instance, we write $\phi_1 \wedge \phi_2 \vee \phi_3$ instead of $((\phi_1 \wedge \phi_2) \vee \phi_3)$ because we assume that negations bind stronger than conjunctions, and conjunctions bind stronger than disjunctions. We also omit double parentheses like in $I_{\Sigma_P}((\phi \wedge \psi))$ and write $I_{\Sigma_P}(\phi \wedge \psi)$ instead.

- We introduce the *material implication* $\phi \Rightarrow \psi$ as an abbreviated form of $\neg\phi \vee \psi$. That is, $\phi \Rightarrow \psi \equiv \neg\phi \vee \psi$. Further, the *material equivalence* $\phi \Leftrightarrow \psi$ indicates on a syntactical level that ϕ implies ψ and ψ implies ϕ, i.e., $\phi \Leftrightarrow \psi \equiv (\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)$.
- We also use the following abbreviations: We write $\phi\psi$ instead of $\phi \wedge \psi$ (juxtaposition indicates conjunction), and we write $\overline{\phi}$ instead of $\neg\phi$ (overline indicates negation).

Propositional logics form the basis of the more complex *relational logics* which are introduced in the next section.

2.2 Relational Logics

Relational logics are *function-free first order logics*[1] defined over finite domains [Genesereth and Kao, 2016; Ullman, 1988]. They extend propositional logics by the concepts of *constants* and *predicates* with which it is possible to formalize statements about single individuals but also relate individuals to each other. With the *quantifiers* $\forall$ and $\exists$, statements about *all* or *some* individuals can be expressed. For example, the natural language statement "all penguins are birds" translates to $\forall X.\mathsf{Penguin}(X) \Rightarrow \mathsf{Bird}(X)$, and "Peter has a friend" to $\exists X.\mathsf{friend}(\mathsf{peter}, X)$. Therewith, relational logics provide expressive languages that can be used to represent and reason about general but also individual-specific knowledge.

In this section, we discuss the general syntax and semantics of relational expressions. Furthermore, we make some specific remarks on the satisfiability and equivalence of relational sentence as well as on counting their models which will be of importance in the further course of the thesis. Again, we loosely follow the explanations in [Beierle and Kern-Isberner, 2014].

▶ Syntax of Relational Formulas

Relational languages $\mathcal{RL}(\Sigma)$ are defined over finite *signatures* Σ which are, in contrast to propositional signatures, tuples that consist of a set of *constants* and a set of *predicates*.

---

[1] Note that in function-free first order logics constants are allowed although they are technically functions of arity 0.

Definition 2.2.1: (Relational) Signature

(cf. e.g., [Beierle and Kern-Isberner, 2014])

A finite *(relational) signature* is a tuple $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ consisting of a finite set of *constants* Const_Σ and a finite set of *predicates* Pred_Σ. A *constant* is a formal representation of an individual or an object,[a] and is typically denoted by a lowercase letter $(a, b, c, \ldots)$. A *predicate* is a *predicate symbol*, here an uppercase letter $(P, Q, R, \ldots)$, together with an *arity* $n \in \mathbb{N}_0$, for example P/n, and formalizes a circumstance (in case of $n = 0$), a concept or a property of an individual ($n = 1$), or a relation between individuals ($n > 1$).

---

[a] We will stick to the term *individual* in the following, regardless of whether we mean a living individual or not.

A *term* of arity $n \in \mathbb{N}_0$ is a tuple of n constants or *(logical) variables*, where variables serve as “placeholders” for constants and are denoted by uppercase letters $X, Y, Z, \ldots$ An *atom* is a predicate P/n together with a term t of the same arity n, denoted by $P(t)$. Atoms are the basic building blocks of the formulas in $\mathcal{RL}(\Sigma)$. An atom the term of which does not mention any variables is called a *ground atom.* The set of all atoms that can be built over Σ is denoted by $\mathsf{Atom}(\Sigma)$, the set of the ground atoms by $\mathsf{grAtom}(\Sigma)$. The set $\mathsf{grAtom}(\Sigma)$ has the cardinality

$$|\mathsf{grAtom}(\Sigma)| = \sum_{P \in \mathsf{Pred}_\Sigma} |\mathsf{Const}_\Sigma|^{\mathsf{arity}(P)},$$

where $\mathsf{arity}(P)$ is the arity of P. We say that a signature Σ' is a *subsignature* of a signature Σ, written $\Sigma' \sqsubseteq \Sigma$, iff $\mathsf{Const}_{\Sigma'} \subseteq \mathsf{Const}_\Sigma$ and $\mathsf{Pred}_{\Sigma'} \subseteq \mathsf{Pred}_\Sigma$. If Σ' is a subsignature of Σ, then this implies $\mathsf{grAtom}(\Sigma') \subseteq \mathsf{grAtom}(\Sigma)$.

Example 2.2.2

We consider three signatures based on which we illustrate the basic notions in this section. Most of the examples in the remainder of the thesis are linked to one of these signatures.

a) We consider a nature watcher who observes that when there is rainy weather but also the sun shines sometimes rainbows occur. In order to formalize this observation, we make use of the nullary predicates ($n = 0$) $\mathsf{Sunshine}/0$, $\mathsf{Rain}/0$, and $\mathsf{Rainbow}/0$ such that the set of predicates of our formalization is

$$\mathsf{Pred}_{\Sigma_{\mathsf{bow}}} = \{\mathsf{Sunshine}/0,\ \mathsf{Rain}/0,\ \mathsf{Rainbow}/0\}.$$

As no constants are needed to represent this scenario, we may set $\mathsf{Const}_{\Sigma_{\mathsf{bow}}} = \emptyset$ and, thus, have $\Sigma_{\mathsf{bow}} = (\emptyset, \mathsf{Pred}_{\mathsf{bow}})$. Note the similarity between this signature and the propositional signature in Example 2.1.3 from Section 2.1. Here, the nullary predicates take the role of the atomic propositions from Example 2.1.3.

b) Our second example deals with the—in the KR community famous—non-flying penguins. Say that an ornithologist observes that most birds are able to fly while she knows that penguins are birds which are not able to fly and, therefore, constitute an exceptional species of birds. Actually, the ornithologist does not know how many birds exist but she definitely knows Tweety which is her domestic penguin. The signature of a formalization of her knowledge could be any $\Sigma_{\mathsf{bfp}} = (\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}, \mathsf{Pred}_{\Sigma_{\mathsf{bfp}}})$ with

$$\mathsf{tweety} \in \mathsf{Const}_{\Sigma_{\mathsf{bfp}}} \quad \text{and} \quad \mathsf{Pred}_{\Sigma_{\mathsf{bfp}}} = \{\mathsf{Bird}/1,\ \mathsf{Flies}/1,\ \mathsf{Penguin}/1\}.$$

We will extend and vary $\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}$ and $\mathsf{Pred}_{\Sigma_{\mathsf{bfp}}}$ in the upcoming examples in order to emphasize different aspects of this scenario, but Σ_{bfp}, as defined here, will always serve as a common basis. It is obvious that $\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}$ should be very large in order to reflect the actual number of birds in the world, in particular compared to $|\{\mathsf{tweety}\}| = 1$.

c) Our last scenario deals with a sociologist who investigates common behavior in peer groups. Among other things, she finds out that friends of smokers usually smoke, too. The formalization of her findings is based on the signature $\Sigma_{\mathsf{smk}} = (\mathsf{Const}_{\Sigma_{\mathsf{smk}}}, \mathsf{Pred}_{\Sigma_{\mathsf{smk}}})$ with

$$\mathsf{peter}, \mathsf{paul}, \mathsf{mary} \in \mathsf{Const}_{\Sigma_{\mathsf{smk}}} \quad \text{and} \quad \mathsf{Pred}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{Friends}/2,\ \mathsf{Smokes}/1\}.$$

Peter, Paul, and Mary are three of her propositi she has interviewed during here case study.

In order to formulate statements in relational logics, for example over the signatures in Example 2.2.2, we employ the notion of *(relational) formulas.*

Definition 2.2.3: (Relational) Formula

(cf. e.g., [Beierle and Kern-Isberner, 2014])

Let Σ be a finite (relational) signature. A *(relational) formula* $\phi \in \mathcal{RL}(\Sigma)$ is either an atom from $\mathsf{Atom}(\Sigma)$, of the form $\top$ *(tautology)* or $\bot$ *(contradiction)*, or a *(compounded) formula*, whereby formulas are compounded by the recursive use of one or more of the logical connectives *negation* $(\neg\phi)$, *conjunction* $(\phi\wedge\psi)$, *dis-*

junction ($\phi \vee \psi$), *material implication* ($\phi \Rightarrow \psi$), *material equivalence* ($\phi \Leftrightarrow \psi$), *universal quantification* ($\forall X.\phi$), and *existential quantification* ($\exists X.\phi$). Hereby, $\phi, \psi \in \mathcal{RL}(\Sigma)$ are formulas, and X is a (logical) variable.

As a *literal*, we understand an atom $A \in \mathsf{Atom}(\Sigma)$ or its negation. If A is a ground atom, then we call the respective literals, A and $\neg A$, *ground literals*. The set of all literals over Σ is denoted by $\mathsf{Lit}(\Sigma)$ while $\mathsf{grLit}(\Sigma)$ denotes the set of all ground literals. With $\mathsf{grAtom}(\phi)$ and $\mathsf{grLit}(\phi)$ we denote the sets of ground atoms and ground literals, respectively, which occur in the formula $\phi \in \mathcal{RL}(\Sigma)$.

Variables in formulas can either be *bounded* by a quantifier or they can be *free*. For example, the variable Y in the formula

$$\phi_{\mathsf{smk}} = \exists Y.(\mathsf{Friends}(X, Y) \wedge \mathsf{Smokes}(Y)), \tag{2.1}$$

representing all individuals X who have at least one smoking friend Y, is bounded by the existential quantification in ϕ_{smk}, while X is out of the scope of any quantification and, therefore, is free. In order to emphasize on the free variables in a formula, we sometimes annotate them as arguments to the formula. For instance, we would write $\phi_{\mathsf{smk}}(X)$ for the formula in (2.1). With $\mathsf{FreeVar}(\phi)$ we denote the set of the free variables in ϕ. For instance, $\mathsf{FreeVar}(\phi_{\mathsf{smk}}) = \{X\}$. Formulas without free variables are used to represent statements that can either be true or false. Therefore, they are called *sentence*.

Definition 2.2.4: Sentence *(cf. e.g., [Beierle and Kern-Isberner, 2014])*

Let Σ be a finite signature, and let $\phi \in \mathcal{RL}(\Sigma)$ be a formula. If $\mathsf{FreeVar}(\phi) = \emptyset$, then ϕ is called *sentence*. The set of all sentences from $\mathcal{RL}(\Sigma)$ is denoted with $\mathcal{RL}^S(\Sigma)$.

By definition, each sentence is also a formula, i.e., $\mathcal{RL}^S(\Sigma) \subseteq \mathcal{RL}(\Sigma)$. Sentences may mention variables, but these variables have to be bounded.

Example 2.2.5

The formula ϕ_{smk} as defined in (2.1) is not a sentence because of its free variable X, i.e., because $\mathsf{FreeVar}(\phi_{\mathsf{smk}}) = \{X\} \neq \emptyset$. In contrast to that, the formula $\phi'_{\mathsf{smk}} \in \mathcal{RL}(\Sigma_{\mathsf{smk}})$ given by

$$\phi'_{\mathsf{smk}} = \forall X.\exists Y.\mathsf{Friends}(X, Y) \wedge \mathsf{Smokes}(Y),$$

"Every person has a friend who smokes."

which is a slight variation of ϕ_{smk} is a sentence because X is now bounded by the universal quantification in ϕ'_{smk}. Consequently, $\mathsf{FreeVar}(\phi'_{\mathsf{smk}}) = \emptyset$ and $\phi'_{\mathsf{smk}} \in \mathcal{RL}^S(\Sigma_{\mathsf{smk}})$.

We call formulas that mention free variables *open*. Open formulas can be made closed (i.e., sentences) by *substituting* the free variables in the formula by constants. The substitution of a free variable $X \in \mathsf{FreeVar}(\phi)$ by a constant $c \in \mathsf{Const}_\Sigma$ in a formula $\phi \in \mathcal{RL}(\Sigma)$ is denoted by $\phi\langle X/c\rangle$ and is the syntactical replacement of every occurrence of the symbol X in ϕ by c. If ϕ mentions more than one free variable, then this substitution process has to be applied to every free variable in ϕ in order to make the formula closed. Note that this intuitive view on substitutions requires that the free variables and the bounded variables in a formula are represented by different symbols. For instance, while the formula $\phi(Y) = A(Y) \wedge \forall X.B(X)$ is suited for substituting Y, the formula $\phi'(X) = A(X) \wedge \forall X.B(X)$ has to be rewritten to $\phi'(Y) = A(Y) \wedge \forall X.B(X)$ before the substitution can be performed. Otherwise, it would not be clear which occurrence of X can be replaced by a constant and which occurrence should not be replaced because it is bounded, at least if the substitution is considered to be a purely syntactic replacement operation as introduced here.

If a sentence $\phi' \in \mathcal{RL}^S(\Sigma)$ originates from the substitution of the free variables in an open formula $\phi \in \mathcal{RL}(\Sigma)$, then ϕ' is called a *(ground) instance* of ϕ. A formula $\phi \in \mathcal{RL}(\Sigma)$ has $|\mathsf{Const}_\Sigma|^{|\mathsf{FreeVar}(\phi)|}$-many instances. The set of all instances of ϕ is denoted by $\mathsf{Inst}_\Sigma(\phi)$. Note that, in fact, $\mathsf{Inst}_\Sigma(\phi)$ does not depend on Σ as a whole but only on Const_Σ. For integrity reasons we keep the subscript Σ, though.

Example 2.2.6

Once again, we consider the formula $\phi_{\mathsf{smk}}(X)$ from (2.1), i.e.,

$$\phi_{\mathsf{smk}}(X) = \exists Y.\mathsf{Friends}(X,Y) \wedge \mathsf{Smokes}(Y).$$

Let $\mathsf{Const}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{peter},\ \mathsf{paul},\ \mathsf{mary}\}$ be the underlying set of constants. Then, the set of the instances of $\phi_{\mathsf{smk}}(X)$ with respect to the signature Σ_{smk} is

$$\begin{aligned}\mathsf{Inst}_{\Sigma_{\mathsf{smk}}}(\phi_{\mathsf{smk}}(X)) = \{&\exists Y.\mathsf{Friends}(\mathsf{peter},Y) \wedge \mathsf{Smokes}(Y),\\ &\exists Y.\mathsf{Friends}(\mathsf{paul},Y) \wedge \mathsf{Smokes}(Y),\\ &\exists Y.\mathsf{Friends}(\mathsf{mary},Y) \wedge \mathsf{Smokes}(Y)\}.\end{aligned}$$

Note that the bounded variable Y is not substituted. The set of the instances of the formula $\phi_{\mathsf{frnd}}(X,Y) = \mathsf{Friends}(X,Y)$ with respect to the signature Σ_{smk} is

$$\mathsf{Inst}_{\Sigma_{\mathsf{smk}}}(\phi_{\mathsf{frnd}}(X,Y)) = \{$$

$$\begin{array}{lll} \mathsf{Friends(peter,peter)}, & \mathsf{Friends(peter,paul)}, & \mathsf{Friends(peter,mary)}, \\ \mathsf{Friends(paul,peter)}, & \mathsf{Friends(paul,paul)}, & \mathsf{Friends(paul,mary)}, \\ \mathsf{Friends(mary,peter)}, & \mathsf{Friends(mary,paul)}, & \mathsf{Friends(mary,mary)} \quad \}. \end{array}$$

The order in which X and Y are substituted in $\phi_{\mathsf{frnd}}(X,Y)$ is irrelevant. This is an observation which also holds in general.

In order to improve the readability of formal expressions, we use the same conventions and abbreviations as in propositional logics (cf. Section 2.1). In addition, we comply with the common precedences of logical connectives,

> negation > conjunction > disjunction > material implication > material equivalence > universal quantification > existential quantification,

where $A > B$ means that A binds stronger than B, and omit parentheses when possible.

▶ Semantics of Sentences

Similar to propositional interpretations, the semantics of sentences in $\mathcal{RL}^S(\Sigma)$ is based on *interpretations* which assign to every sentence a truth value from $\{0,1\}$, where 1 means *true* and 0 means *false*. Formulas which mention free variables have to be made closed by a proper instantiation before they can be interpreted. Note that our view on the semantics of $\mathcal{RL}(\Sigma)$-expressions follows Jacques Herbrand's idea of a "semantics through syntax," also known as *Herbrand semantics* (cf. [Herbrand, 1930]). The basic idea is that constants are representatives of themselves and interpretations are fully determined by the truth assignments that are made to the ground atoms in $\mathsf{grAtom}(\Sigma)$. For alternative semantics of more general first-order expressions, please see [Ebbinghaus et al., 2021].

Definition 2.2.7: Truth Assignment and (Relational) Interpretation
(cf. e.g., [Beierle and Kern-Isberner, 2014])

Let Σ be a finite signature. A mapping $\theta\colon \mathsf{grAtom}(\Sigma) \to \{0,1\}$ which assigns a truth value from $\{0,1\}$, to every ground atom from $\mathsf{grAtom}(\Sigma)$ is called a *truth assignment*. With $\Theta(\Sigma)$ we denote the set of all truth assignments over Σ. Truth assignments $\theta \in \Theta(\Sigma)$ are extended to *(relational) interpretations* $I_\theta\colon \mathcal{RL}^S(\Sigma) \to \{0,1\}$ by the recursive evaluation of the logical connectives in $\mathcal{RL}(\Sigma)$ as shown in Table 2.2.

Sentence	Interpretation
A	$I_\theta(A) = \theta(A)$
$\top$	$I_\theta(\top) = 1$
$\bot$	$I_\theta(\bot) = 0$
$\neg\phi$	$I_\theta(\neg\phi) = \begin{cases} 1 & \text{if} \quad I_\theta(\phi) = 0 \\ 0 & \text{otherwise} \end{cases}$
$\phi \wedge \psi$	$I_\theta(\phi \wedge \psi) = \begin{cases} 1 & \text{if} \quad I_\theta(\phi) = 1 \quad \text{and} \quad I_\theta(\psi) = 1 \\ 0 & \text{otherwise} \end{cases}$
$\phi \vee \psi$	$I_\theta(\phi \vee \psi) = \begin{cases} 1 & \text{if} \quad I_\theta(\phi) = 1 \quad \text{or} \quad I_\theta(\psi) = 1 \\ 0 & \text{otherwise} \end{cases}$
$\phi \Rightarrow \psi$	$I_\theta(\phi \Rightarrow \psi) = \begin{cases} 1 & \text{if} \quad I_\theta(\phi) = 0 \quad \text{or} \quad I_\theta(\psi) = 1 \\ 0 & \text{otherwise} \end{cases}$
$\phi \Leftrightarrow \psi$	$I_\theta(\phi \Leftrightarrow \psi) = \begin{cases} 1 & \text{if} \quad I_\theta(\phi) = I_\theta(\psi) \\ 0 & \text{otherwise} \end{cases}$
$\forall X.\gamma$	$I_\theta(\forall X.\gamma) = \begin{cases} 1 & \text{if} \quad \forall c \in \mathsf{Const}_\Sigma \colon I_\theta(\gamma\langle X/c\rangle) = 1 \\ 0 & \text{otherwise} \end{cases}$
$\exists X.\gamma$	$I_\theta(\exists X.\gamma) = \begin{cases} 1 & \text{if} \quad \exists c \in \mathsf{Const}_\Sigma \colon I_\theta(\gamma\langle X/c\rangle) = 1 \\ 0 & \text{otherwise} \end{cases}$

Table 2.2: Recursive interpretation of sentences in $\mathcal{RL}^S(\Sigma)$. Here, $A \in \mathsf{grAtom}(\Sigma)$ is a ground atom, $\phi, \psi \in \mathcal{RL}^S(\Sigma)$ are sentences, $\gamma \in \mathcal{RL}(\Sigma)$ is a formula with $\mathsf{FreeVar}(\gamma) = \{X\}$, and $\theta \in \Theta(\Sigma)$ is a truth assignment.

The set of all interpretations over the signature Σ is denoted by $\mathsf{Int}(\Sigma)$. In total there are $2^{|\mathsf{grAtom}(\Sigma)|}$-many interpretations in $\mathsf{Int}(\Sigma)$ because there are $2^{|\mathsf{grAtom}(\Sigma)|}$-many possibilities of assigning a truth value to the ground atoms in $\mathsf{grAtom}(\Sigma)$. Hence, the number of different interpretations is exponential in the number of different ground atoms in $\mathsf{grAtom}(\Sigma)$.

An interpretation $I \in \mathsf{Int}(\Sigma)$ is a Σ-*model* of a sentence $\phi \in \mathcal{RL}^S(\Sigma)$ if $I(\phi) = 1$. We also say that ϕ is Σ-*satisfied* by I. The set of all Σ-models of ϕ is denoted by $\mathsf{Mod}_\Sigma(\phi)$. If $\mathsf{Mod}_\Sigma(\phi) \neq \emptyset$, then we say that ϕ is Σ-*satisfiable*. We will see later on in Example 2.2.10 that, indeed, the satisfiability of a sentence depends on the signature Σ, which is why we say Σ-*model* unless the signature is known from the context. Then, we also use the term *model* for simplicity.

A sentence $\phi \in \mathcal{RL}^S(\Sigma)$ Σ-*entails* a sentence ψ, written $\phi \models_\Sigma \psi$, iff $\mathsf{Mod}_\Sigma(\phi) \subseteq \mathsf{Mod}_\Sigma(\psi)$. Iff ϕ and ψ Σ-entail each other, i.e., iff $\mathsf{Mod}_\Sigma(\phi) = \mathsf{Mod}_\Sigma(\psi)$, then ϕ and ψ are *(logically)* Σ-*equivalent*, denoted by $\phi \equiv_\Sigma \psi$. Also here, we remove the signature Σ from terms and symbols if it is clear from the context which signature is used.

▶ Remarks on Satisfiability, Models, and Counting

We make some specific remarks on the satisfiability of sentences, as well as on counting their models. In particular, we will give an example which illustrates the recursive interpretation of sentences later on. Beforehand, we want to point to the fact that under certain circumstances the interpretation of conjunctions is related to multiplication and the interpretation of disjunctions is related to addition. This insight will be of importance when we aim at counting the number of models of sentences in Section 6.1, i.e., when calculating $|\mathsf{Mod}_\Sigma(\phi)|$ for a sentence ϕ.

We call sentences $\phi, \psi \in \mathcal{RL}^S(\Sigma)$ *mutually exclusive in* $I \in \mathsf{Int}(\Sigma)$ iff $I(\phi) \neq I(\psi)$. Further, ϕ and ψ are *mutually exclusive* iff $I(\phi) \neq I(\psi)$ holds for all $I \in \mathsf{Int}(\Sigma)$.

Proposition 2.2.8: Properties of Interpretations

Let Σ be a finite signature, let $\phi, \psi \in \mathcal{RL}^S(\Sigma)$ be sentences, and let $I \in \mathsf{Int}(\Sigma)$ be an interpretation. Then,

- $I(\phi \wedge \psi) = I(\phi) \cdot I(\psi)$,
- $I(\phi \vee \psi) = I(\phi) + I(\psi) = 1$ if ϕ and ψ are *mutually exclusive in* I.

Proof. If $I(\phi) = I(\psi) = 1$, then $I(\phi \wedge \psi) = 1 = 1 \cdot 1 = I(\phi) \cdot I(\psi)$. In all other cases, $I(\phi \wedge \psi) = 0$ as $I(\phi) \cdot I(\psi) = 0$. The latter holds because at least one factor is zero, then. Further, if ϕ and ψ are mutually exclusive in I, then either $I(\phi) = 1$ and $I(\psi) = 0$, or $I(\phi) = 0$ and $I(\psi) = 1$ so that $I(\phi \vee \psi) = 1 = I(\phi) + I(\psi)$. □

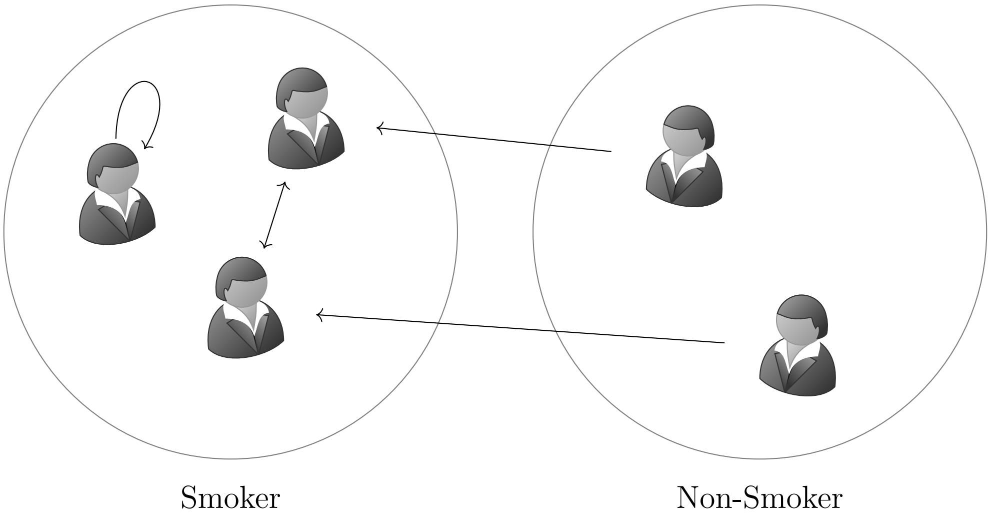

Figure 2.1: Depiction of the scenario in Example 2.2.9.b. There are two groups of persons, smokers and non-smokers. In order that the sentence ϕ'_{smk} is satisfied, each person has to have a smoker as a friend (exemplarily visualized by the arrows). All other friendship relations are optional.

Proposition 2.2.8 constitutes the formal basis of fundamental model counting techniques (cf. Section 6.1) and allows one to understand the values 0 and 1 as numbers rather than truth values.

Due to the finiteness of Σ, quantifications are Σ-equivalent to appropriate conjunctions or disjunctions. For $\phi \in \mathcal{RL}(\Sigma)$ with $X \in \mathsf{FreeVar}(\phi)$, we have

- $\forall X.\phi \equiv_\Sigma \bigwedge_{c \in \mathsf{Const}_\Sigma} \phi\langle X/c\rangle$,
- $\exists X.\phi \equiv_\Sigma \bigvee_{c \in \mathsf{Const}_\Sigma} \phi\langle X/c\rangle$,

which is why relational logics are sometimes defined without quantifiers. However, quantification plays an important role in efficient model counting because it implies implicit information about symmetries between constants which can be used to apply combinatorial arguments. We give an Example and continue this line of thought in Section 6.1.

Example 2.2.9

We consider the sentence

$$\phi'_{\mathsf{smk}} = \forall X.\exists Y.\mathsf{Friends}(X,Y) \wedge \mathsf{Smokes}(Y),$$

"Every person has a friend who smokes."

whereby the set of predicates is fixed, $\mathsf{Pred}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{Friends}/2, \mathsf{Smokes}/1\}$, and we will vary the set of constants $\mathsf{Const}_{\Sigma_{\mathsf{smk}}}$.

a) Let $\mathsf{Const}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{peter},\ \mathsf{paul},\ \mathsf{mary}\}$, and let $I_\theta \in \mathsf{Int}(\Sigma_{\mathsf{smk}})$ be the interpretation determined by the truth assignment

$$\begin{aligned} \theta(\mathsf{Friends}(X,Y)) &= 1, && \text{for } X,Y \in \{\mathsf{peter},\mathsf{paul},\mathsf{mary}\},\ X \neq Y,\\ \theta(\mathsf{Friends}(X,X)) &= 0, && \text{for } X \in \{\mathsf{peter},\mathsf{paul},\mathsf{mary}\},\\ \theta(\mathsf{Smokes}(\mathsf{peter})) &= 1, &&\\ \theta(\mathsf{Smokes}(X)) &= 0, && \text{for } X \in \{\mathsf{paul},\mathsf{mary}\}. \end{aligned}$$

> “Peter Paul and Mary are friends. While Peter is a smoker, Paul and Mary do not smoke.”

Then, according to the equivalence between (finite) universal quantifications and conjunctions, we obtain

$$\begin{aligned} I_\theta(\phi'_{\mathsf{smk}}) &= I_\theta(\forall X.\exists Y.\mathsf{Friends}(X,Y) \wedge \mathsf{Smokes}(Y))\\ &= I_\theta((\exists Y.\mathsf{Friends}(\mathsf{peter},Y) \wedge \mathsf{Smokes}(Y))\\ &\qquad \wedge (\exists Y.\mathsf{Friends}(\mathsf{paul},Y) \wedge \mathsf{Smokes}(Y))\\ &\qquad \wedge (\exists Y.\mathsf{Friends}(\mathsf{mary},Y) \wedge \mathsf{Smokes}(Y))). \end{aligned}$$

By applying Proposition 2.2.8 and executing the existential quantification in ϕ'_{smk}, it follows that

$$\begin{aligned} I_\theta(\phi'_{\mathsf{smk}}) = I_\theta(&\mathsf{Friends}(\mathsf{peter},\mathsf{peter}) \wedge \mathsf{Smokes}(\mathsf{peter})\\ &\vee \mathsf{Friends}(\mathsf{peter},\mathsf{paul}) \wedge \mathsf{Smokes}(\mathsf{paul}))\\ &\vee \mathsf{Friends}(\mathsf{peter},\mathsf{mary}) \wedge \mathsf{Smokes}(\mathsf{mary}))\\ \cdot\, I_\theta(&\mathsf{Friends}(\mathsf{paul},\mathsf{peter}) \wedge \mathsf{Smokes}(\mathsf{peter})\\ &\vee \mathsf{Friends}(\mathsf{paul},\mathsf{paul}) \wedge \mathsf{Smokes}(\mathsf{paul}))\\ &\vee \mathsf{Friends}(\mathsf{paul},\mathsf{mary}) \wedge \mathsf{Smokes}(\mathsf{mary}))\\ \cdot\, I_\theta(&\mathsf{Friends}(\mathsf{mary},\mathsf{peter}) \wedge \mathsf{Smokes}(\mathsf{peter})\\ &\vee \mathsf{Friends}(\mathsf{mary},\mathsf{paul}) \wedge \mathsf{Smokes}(\mathsf{paul}))\\ &\vee \mathsf{Friends}(\mathsf{mary},\mathsf{mary}) \wedge \mathsf{Smokes}(\mathsf{mary})). \end{aligned}$$

We observe that

$$\begin{aligned} I_\theta(&\mathsf{Friends}(\mathsf{peter},\mathsf{peter}) \wedge \mathsf{Smokes}(\mathsf{peter})\\ &\vee \mathsf{Friends}(\mathsf{peter},\mathsf{paul}) \wedge \mathsf{Smokes}(\mathsf{paul})) \end{aligned}$$

$$\vee\ \mathsf{Friends}(\mathsf{peter}, \mathsf{mary}) \wedge \mathsf{Smokes}(\mathsf{mary})) = 0$$

holds because

$$\theta(\mathsf{Friends}(\mathsf{peter}, \mathsf{peter})) = \mathsf{Smokes}(\mathsf{paul}) = \mathsf{Smokes}(\mathsf{mary}) = 0$$

and, hence, each of the disjuncts in this subsentence of ϕ'_{smk} is interpreted to 0 (false). The other subsentences of ϕ'_{smk} are evaluated to 1 (true), because Peter is a smoker in I_θ and on friendly terms with Paul and Mary, so that altogether,

$$I_\theta(\phi'_{\mathsf{smk}}) = 0 \cdot 1 \cdot 1 = 0.$$

In conclusion, the sentence ϕ'_{smk} is false in I_θ because Peter does not have smoking friends in I_θ.

Note that in case of $|\mathsf{Const}_\Sigma| = 3$ the sentence ϕ'_{smk} has $1,183$ models already, where the number of all interpretations in $\mathsf{Int}(\Sigma)$ is $4,096$. It becomes clear that if one is interested in the model count of sentences like ϕ'_{smk} one needs sophisticated model counting strategies, in particular if $|\mathsf{Const}_\Sigma|$ is much larger and, hence, closer to realistic domain sizes. In the next part of this example, we demonstrate such model counting strategies.

b) Now, let $\mathsf{Const}_{\mathsf{smk}}$ be an arbitrary set of constants of finite size $k = |\mathsf{Const}_{\mathsf{smk}}|$ with $k \geq 1$. We illustrate the importance of symmetries and of indistinguishabilities between constants conveyed by quantifiers when counting the number of models of ϕ'_{smk}. First, we note that, in general, the relation of being friends formalized by the binary predicate $\mathsf{Friends}/2$ is not necessarily symmetric or non-reflexive. That is, being friends is a directed relation here, and every person represented by a constant in $\mathsf{Const}_{\mathsf{smk}}$ can be friends with every other person, also with themselves.

In order to count the models of ϕ'_{smk}, we divide the set of constants $\mathsf{Const}_{\mathsf{smk}}$ into two disjoint sets, namely into the sets containing those constants which represent smokers and those which represent non-smokers. Say that there are s-many smokers the number of which may range from 0 to k. Then, there are $(k-s)$-many non-smokers. In general, every constant could represent a smoker so that there are $\binom{k}{s}$ ways of selecting the s-many constants c for which $\mathsf{Smokes}(c)$ is true among the k-many constants. For the remaining constants c', $\neg\mathsf{Smokes}(c')$ holds.

An interpretation $I \in \mathsf{Int}(\Sigma_{\mathsf{smk}})$ satisfies ϕ'_{smk}, if for every constant $c \in \mathsf{Const}_{\Sigma_{\mathsf{smk}}}$ there is a constant $c' \in \mathsf{Const}_{\Sigma_{\mathsf{smk}}}$ which represents a smoking friend of c so that both $\mathsf{Friends}(c, c')$ and $\mathsf{Smokes}(c')$ is true. This holds regardless whether c itself is representing a smoker or not because the atom $\mathsf{Smokes}(Y)$ refers to the second argument of $\mathsf{Friends}(X, Y)$, in our case the constant c'. That is, all constants c behave interchangeably with respect to that requirement of having a smoking friend. In addition, if a smoking friend c' of c exists, whether c is Friends-related to other constants which represent smokers or to any constant which represents a non-smoker is irrelevant. This scenario is depicted in Figure 2.1. Thus, in total there are $(2^s - 1) \cdot (2^{k-s})$-many possible ways in which the sentence $\exists Y.\mathsf{Friends}(c, Y) \wedge \mathsf{Smokes}(Y)$ can be satisfied. Hereby, the factor 2^{k-s} is the number of all possible Friends-relations of c to non-smokers: With each of the $(k-s)$-many non-smokers c can either be Friends-related or not. And the factor $2^s - 1$ reflects the admissible Friends-relations of c to smokers: There are s-many smokers with which c can be Friends-related or not. However, in the case where c is not Friends-related to any of the smokers, the sentence $\exists Y.\mathsf{Friends}(c, Y) \wedge \mathsf{Smokes}(Y)$ is not satisfied. Hence, we have to subtract this one case from the 2^s-many combinations of "being or not being friends with one of the smokers" which results in the factor $2^s - 1$.

Because these ideas on evaluating $\exists Y.\mathsf{Friends}(c, Y) \wedge \mathsf{Smokes}(Y)$ hold for all constants $c \in \mathsf{Const}_{\Sigma_{\mathsf{smk}}}$, we have to multiply the numbers of arranging the friendship relation between smokers and non-smokers k-times and obtain in total

$$|\mathsf{Mod}_{\Sigma_{\mathsf{smk}}}(\phi'_{\mathsf{smk}})| = \sum_{s=0}^{k} \binom{k}{s} \left((2^s - 1) \cdot (2^{k-s})\right)^k .$$

For instance, for $k = 1$ this formula yields 1 model of ψ'_{smk} out of 4 interpretations. For $k = 2$ we get 17 models out of 64 interpretations, and for $k = 3$ we obtain the already mentioned $1,183$ models where the number of interpretations is $4,096$.

As mentioned above, the satisfiability of sentences depends on the signature Σ, more precisely on the size of Const_{Σ}, as the following example shows.

Example 2.2.10
We consider the sentence

$$\phi = \exists X.\ A(X) \wedge \exists Y.\ \overline{A(Y)}$$

which is not Σ-satisfiable if $|\mathsf{Const}_\Sigma| = 1$. To see this, let $\mathsf{Const}_\Sigma = \{c\}$ without loss of generality. Then, no interpretation $I \in \mathsf{Int}(\Sigma)$ can fulfill $I(A(c)) = 1$ and $I(\overline{A(c)}) = 1$ at the same time and, thus, at least one conjunct in ϕ is false. On the other hand, if $|\mathsf{Const}_\Sigma| > 1$, there are at least two constants in Const_Σ, say c and d, and we can find an interpretation $I \in \mathsf{Int}(\Sigma)$ with $I(A(c)) = 1$ and $I(\overline{A(d)}) = 1$ which is consequently a Σ-model of ϕ.

While the satisfiability of sentences in general first-order logics is undecidable [Tarski et al., 2010], the question whether a sentence from the fragment $\mathcal{RL}^S(\Sigma)$ can be satisfied is decidable thanks to the finiteness of Σ. Every sentence in $\mathcal{RL}^S(\Sigma)$ is either Σ-satisfiable or Σ-equivalent to $\bot$. Checking whether two sentences $\phi, \psi \in \mathcal{RL}^S(\Sigma)$ are Σ-equivalent is equivalent to checking the (non-)Σ-satisfiability of the sentence $\phi\bar{\psi} \vee \bar{\phi}\psi$.

As in the propositional case (cf. Section 2.1), we assume that the reader is familiar with the common logical rules for rearranging sentences which preserve equivalence. Besides the distributive law, one further example of such a rule is *idempotence* which states that conjunctions (disjunctions) that mention the same conjunct (disjunct) ϕ twice can be reduced to ϕ.

Proposition 2.2.11: Idempotence of Conjunctions and Disjunctions
Let Σ be a finite signature, let $\phi \in \mathcal{RL}^S(\Sigma)$ be a sentence, and let $I \in \mathsf{Int}(\Sigma)$ be an interpretation. Then,

$$\phi \wedge \phi \equiv_\Sigma \phi, \qquad \text{and} \qquad \phi \vee \phi \equiv_\Sigma \phi.$$

Proof. This proposition directly follows from the definition of interpretations. □

We emphasize the idempotence here because we will revisit this rule in Section 6.2 when we consider so-called *structured sentences*. For structured sentences, idempotence will not hold in general.

Note that the statements about Σ-models which we have made in this section apply mutatis mutandis to sets of sentences $\Phi \subseteq \mathcal{RL}^S(\Sigma)$, too. In particular, we have

$$\mathsf{Mod}_\Sigma(\Phi) = \{I \in \mathsf{Int}(\Sigma) \mid \forall \phi \in \Phi \colon I(\phi) = 1\}$$

for the models of sets of sentences. Therewith, $\Phi \models_\Sigma \Psi$ iff $\mathsf{Mod}_\Sigma(\Phi) \subseteq \mathsf{Mod}_\Sigma(\Psi)$ follows for $\Phi, \Psi \subseteq \mathcal{RL}^S(\Sigma)$. Also in analogy, a set of sentences Φ is Σ-*consistent* iff $\mathsf{Mod}_\Sigma(\Phi) \neq \emptyset$. Note that we omit the set braces of unit sets if this does not cause ambiguities. For example, we write $\mathsf{Mod}_\Sigma(\phi)$ instead of $\mathsf{Mod}_\Sigma(\Phi)$ if $\Phi = \{\phi\}$.

2.3 Relational Logics With Domain and Set Constraints

In this section, we extend relational logics $\mathcal{RL}(\Sigma)$ by additional connectives which make use of so-called *domain constraints* or *set constraints*. Note that we will stick to the notation $\mathcal{RL}(\Sigma)$ for relational logics with these additional connectives. With domain and set constraints, we restrict the scope of quantification (and, later on, also the scope of so-called open conditionals), similar to the domain constraints in [Van den Broeck, 2013] and the many-sorted signatures and constraint formulas in [Fisseler, 2010]. Introducing such domain and set constraints allows for a more concise knowledge representation and more flexibility when rearranging sentences but does not affect the expressivity of the language. Domain and set constraints are of importance in the context of first-order (typed) model counting (cf. Chapter 6) as they allow for more efficient counting techniques.

▶ Domain Constraints

First, we introduce quantifiers with domain constraints.

Definition 2.3.1: Quantifiers with Domain Constraints

(based on [Van den Broeck, 2013])

Let $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ be a finite signature, let $a \in \mathsf{Const}_\Sigma$ be a constant, let $\mathcal{C} \subseteq \mathsf{Const}_\Sigma$ be a set of constants, and let X and Y be logical variables. Then, a *domain constraint* for X is an expression of the form

$$X \neq a, \quad X \neq Y, \quad X \in \mathcal{C}, \quad \text{or} \quad X \in \mathsf{Const}_\Sigma \setminus \mathcal{C}.$$

In addition, let $\phi \in \mathcal{RL}(\Sigma)$ be a formula. Then,

- $\forall_{\mathsf{CS}} X.\phi$ is a *universal quantification with domain constraints*,
- $\exists_{\mathsf{CS}} X.\phi$ is an *existential quantification with domain constraints*,

iff CS is a finite set of domain constraints for X such that for every domain constraint of the form $X \neq Y$ in CS it holds that $Y \in \mathsf{FreeVar}(\phi)$.

The quantifiers with domain constraints $\forall_{\mathsf{CS}} X.\phi$ and $\exists_{\mathsf{CS}} X.\phi$ restrict the scope of the quantification to the solutions of the domain constraints in CS. From now on, we assume that $\mathcal{RL}(\Sigma)$ is equipped with these quantifiers with domain constraints as well.

Example 2.3.2
We consider the sentence

$$\phi''_{\mathsf{smk}} = \forall X.\exists_{Y \neq X} Y.\mathsf{Friends}(X,Y) \wedge \mathsf{Smokes}(Y)$$

which is a more adequate representation of the statement that "every person has a friend who smokes" than the sentence

$$\phi'_{\mathsf{smk}} = \forall X.\exists Y.\mathsf{Friends}(X,Y) \wedge \mathsf{Smokes}(Y)$$

from Example 2.2.5 because it excludes the case in which a person is a friend of herself. The inequation $Y \neq X$ is a proper domain constraint for Y because X is a free variable in the subformula $\mathsf{Friends}(X,Y) \wedge \mathsf{Smokes}(Y)$.

The formal interpretation of a quantifier with domain constraints, either $\forall_{\mathsf{CS}} X.\phi$ or $\exists_{\mathsf{CS}} X.\phi$, requires the evaluation of the domain constraints in CS which restrict the scope of the quantification. For this, we define the *solution set* $\mathsf{Sol}_\Sigma(\mathsf{CS})$ of a set of domain constraints CS. Hereby, we may stick to the case in which CS does not mention domain constraints of the form $X \neq Y$ because we can reduce such constraints to constraints of the form $X \neq a$ for constants $a \in \mathsf{Const}_\Sigma$ as we will see later on.

Definition 2.3.3: Solution Set of Domain Constraints
(based on [Van den Broeck, 2013])

Let $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ be a finite signature, and let CS be a set of domain constraints for the variable X without domain constraints of the form $X \neq Y$. Then, we define the *solution set* of CS with respect to Σ by

$$\begin{aligned}\mathsf{Sol}_\Sigma(\mathsf{CS}) = \{c \in \mathsf{Const}_\Sigma \mid\ & \forall (X \neq a) \in \mathsf{CS}\colon c \neq a,\\ & \forall (X \in \mathcal{C}) \in \mathsf{CS}\colon c \in \mathcal{C},\\ & \forall (X \in \mathsf{Const}_\Sigma \setminus \mathcal{C}) \in \mathsf{CS}\colon c \in \mathsf{Const}_\Sigma \setminus \mathcal{C}\}.\end{aligned}$$

Based on the notion of solution sets of domain constraints, we can extend the recursive interpretation of sentences in $\mathcal{RL}^S(\Sigma)$ to quantifiers with domain constraints.

Definition 2.3.4: Interpretation of Quantifiers with Domain Constraints *(based on [Van den Broeck, 2013])*

Let Σ be a finite signature, let $\phi \in \mathcal{RL}(\Sigma)$ be a formula with $\mathsf{FreeVar}(\phi) = \{X\}$, let CS be a set of domain constraints for X without domain constraints of the form $X \neq Y$, and let $I \in \mathsf{Int}(\Sigma)$ be an interpretation. Then, we define

- $I(\forall_{\mathsf{CS}} X.\phi) = \begin{cases} 1 & \text{if} \quad \forall c \in \mathsf{Sol}_\Sigma(\mathsf{CS})\colon I(\phi\langle X/c\rangle) = 1 \\ 0 & \text{otherwise} \end{cases}$,
- $I(\exists_{\mathsf{CS}} X.\phi) = \begin{cases} 1 & \text{if} \quad \exists c \in \mathsf{Sol}_\Sigma(\mathsf{CS})\colon I(\phi\langle X/c\rangle) = 1 \\ 0 & \text{otherwise} \end{cases}$.

Note that the substitution $\langle X/c\rangle$ in $\phi\langle X/c\rangle$ means that the variable X is replaced by the constant c in every domain constraint which is mentioned in ϕ, too.

We now discuss why it is no problem to exclude domain constraints of the form $X \neq Y$ from the set of domain constraints CS in Definition 2.3.4. In other words, the extension of interpretations to sentences which mention quantifiers with domain constraints as in Definition 2.3.4 is well-defined even if the sentences mention domain constraints of the form $X \neq Y$. The reason for this is that the outermost quantification in a sentence $\psi \in \mathcal{RL}^S(\Sigma)$ is always of the form $\forall_{\mathsf{CS}} X.\phi$ or $\exists_{\mathsf{CS}} X.\phi$ with $\mathsf{FreeVar}(\phi) \subseteq \{X\}$. Otherwise, there would be another free variable in ϕ that is unbounded in ψ which contradicts the fact that ψ is a sentence. Hence, we are in the situation where CS cannot mention a domain constraint of the form $X \neq Y$ because there is no such free variable Y.

With this finding, a sentence ψ can be interpreted recursively by beginning with the outermost connective. When interpreting the outermost quantification, say $\forall_{\mathsf{CS}} X.\phi$ (the case $\exists_{\mathsf{CS}} X.\phi$ is analogous), the variable X is substituted by a constant $c \in \mathsf{Sol}_\Sigma(\mathsf{CS})$ and the formula ϕ is made closed (i.e., ϕ is replaced by $\phi\langle X/c\rangle$, cf. Definition 2.3.4). In particular, any domain constraint of the form $Y \neq X$ which is mentioned in ϕ is replaced by $Y \neq c$. Then, the outermost quantification with domain constraints in $\phi\langle X/c\rangle$ is of the form $\forall_{\mathsf{CS}} Y.\phi'$ or $\exists_{\mathsf{CS}} Y.\phi'$ where CS does not mention a domain constraint of the form $Y \neq X$ again, and so on. Thus, every sentence in $\mathcal{RL}^S(\Sigma)$ can be interpreted by a recursive application of the rules in Table 2.2 and Definition 2.3.4 when beginning with the interpretation of the outermost connective.

Example 2.3.5

Let $\mathsf{Const}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{peter}, \mathsf{paul}, \mathsf{mary}\}$. We consider the sentence

$$\phi'''_{\mathsf{smk}} = \forall_{X \in \{\mathsf{paul}, \mathsf{mary}\}} X . \exists_{Y \neq X} . \mathsf{Friends}(X, Y) \wedge \mathsf{Smokes}(Y)$$

"Paul and Mary have a friend who smokes."

and any truth assignment $\theta \in \Theta(\Sigma_{\mathsf{smk}})$ with (cf. Example 2.2.9)

$$\begin{aligned} \theta(\mathsf{Friends}(X, Y)) &= 1 && \forall X, Y \in \{\mathsf{peter}, \mathsf{paul}, \mathsf{mary}\}, X \neq Y, \\ \theta(\mathsf{Smokes}(\mathsf{peter})) &= 1, \\ \theta(\mathsf{Smokes}(X)) &= 0 && \forall X \in \{\mathsf{paul}, \mathsf{mary}\}. \end{aligned}$$

Note that the interpretation of $\mathsf{Friends}(X, X)$ for $X \in \mathsf{Const}_\Sigma$ does not play a role here. Then,

$$\begin{aligned} I_\theta(\phi'''_{\mathsf{smk}}) &= I_\theta(\forall_{\{X \in \{\mathsf{paul}, \mathsf{mary}\}\}} X . \exists_{\{Y \neq X\}} . \mathsf{Friends}(X, Y) \wedge \mathsf{Smokes}(Y)) \\ &= I_\theta(\exists_{\{Y \neq \mathsf{paul}\}} Y . \mathsf{Friends}(\mathsf{paul}, Y) \wedge \mathsf{Smokes}(Y)) \\ &\quad \cdot I_\theta(\exists_{\{Y \neq \mathsf{mary}\}} Y . \mathsf{Friends}(\mathsf{mary}, Y) \wedge \mathsf{Smokes}(Y)) \\ &= I_\theta(\mathsf{Friends}(\mathsf{paul}, \mathsf{peter}) \wedge \mathsf{Smokes}(\mathsf{peter}) \\ &\qquad \vee \mathsf{Friends}(\mathsf{paul}, \mathsf{mary}) \wedge \mathsf{Smokes}(\mathsf{mary})) \\ &\quad \cdot I_\theta(\mathsf{Friends}(\mathsf{mary}, \mathsf{peter}) \wedge \mathsf{Smokes}(\mathsf{peter}) \\ &\qquad \vee \mathsf{Friends}(\mathsf{mary}, \mathsf{paul}) \wedge \mathsf{Smokes}(\mathsf{paul})) \\ &= 1 \cdot 1 = 1. \end{aligned}$$

We give some additional remarks on quantifiers with domain constraints:

- Quantifiers without domain constraints can be mimicked by quantifiers with domain constraints by setting $\mathsf{CS} = \emptyset$. Then, there is no constraint which restricts the scope of the quantification and we have, for $\phi \in \mathcal{RL}(\Sigma)$,

  $$\forall_\emptyset X . \phi \equiv \forall X . \phi \quad \text{and} \quad \exists_\emptyset X . \phi \equiv \exists X . \phi .$$

  In particular, $\forall_\emptyset X . \phi$ resp. $\exists_\emptyset X . \phi$ does not mean that the quantification ranges over the empty set but over all constants from Const_Σ.

- If the set of domain constraints CS contains only a single domain constraint, i.e., if $\mathsf{CS} = \{C\}$ for a domain constraint C, then we sometimes omit the set braces and write C instead of $\{C\}$ to shorten expressions.

- Like quantifiers without domain constraints, quantifiers with domain constraints are equivalent to conjunctions or disjunctions due to the finiteness of Σ:

$$\forall_{\mathsf{CS}} X.\phi \equiv \bigwedge_{c \in \mathsf{Sol}_{\Sigma}(\mathsf{CS})} \phi\langle X/c\rangle \quad \text{and} \quad \exists_{\mathsf{CS}} X.\phi \equiv \bigvee_{c \in \mathsf{Sol}_{\Sigma}(\mathsf{CS})} \phi\langle X/c\rangle.$$

- If several universal quantifiers immediately follow each other like in the formula $\forall_{\mathsf{CS}_1} X_1 \ldots \forall_{\mathsf{CS}_m} X_m.\phi$, then we write $\forall_{\mathsf{CS}}\{X_1, \ldots, X_m\}.\phi$ with $\mathsf{CS} = \bigcup_{i=1}^{m} \mathsf{CS}_m$ for short. In the same way we abbreviate successive existential quantifiers $\exists_{\mathsf{CS}_1} X_1 \ldots \exists_{\mathsf{CS}_m} X_m.\phi$ with $\exists_{\mathsf{CS}}\{X_1, \ldots, X_m\}.\phi$. Note that this abbreviated form of writing multiple quantifiers in set notation ignores the order in which the quantifiers are executed. This, however, is unproblematic because of the commutativity of universal and existential quantification and the commutativity of domain constraints, especially of those domain constraints of the form $X \neq Y$.

We give an example where we use the abbreviated form of two successive existential quantifiers.

Example 2.3.6
The sentence

$$\phi''''_{\mathsf{smk}} = \exists_{\emptyset} X.\exists_{\{Y \neq X\}} Y.\mathsf{Friends}(X, Y) \wedge \neg\mathsf{Smokes}(Y)$$

“There is someone who has a non-smoking friend.”

can be rewritten as

$$\phi''''_{\mathsf{abbr}} = \exists_{Y \neq X}\{X, Y\}.\ \mathsf{Friends}(X, Y)\overline{\mathsf{Smokes}(Y)}.$$

Note that whether the existential quantification is executed with respect to the variable X or with respect to the variable Y first does not affect the formal interpretation of the sentence ϕ''''_{abbr}. In particular, it is irrelevant whether the domain constraint $Y \neq X$ applies to the variable Y or to X.

Recall that, among others, we allow for domain constraints of the form $X \in \mathcal{C}$ resp. $X \in \mathsf{Const}_{\Sigma} \setminus \mathcal{C}$ where $\mathcal{C}$ is a set of constants. Sometimes it is useful to vary this set $\mathcal{C}$. Actually, on an informal level, this has happened in Example 2.2.9 where we varied the set of smokers resp. the set of non-smokers. For this reason, in the next paragraph, we introduce the concepts of *set variables* and *set constraints* which are formal realizations of the idea of varying sets.

▶ Set Constraints

In the previous paragraph about domain constraints, we have introduced domain constraints of the form $X \in \mathcal{C}$ resp. $X \in \mathsf{Const}_\Sigma \setminus \mathcal{C}$ where $\mathcal{C} \subseteq \mathsf{Const}_\Sigma$ is a fixed set of constants. Here, we treat the symbol $\mathcal{C}$ in domain constraints as a *variable* and iterate over Const_Σ. We call variables which represent sets of constants *set variables* and realize the iteration process by so-called *set quantifications*.

Definition 2.3.7: Set Quantification *(based on [Van den Broeck, 2013])*

Let $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ be a finite signature, let $\phi \in \mathcal{RL}(\Sigma)$ be a formula, let $\mathcal{D}$ be a subset of Const_Σ, and let $\mathcal{C}$ be a *set variable*. Then, an expression of the form $\mathcal{C} \subseteq \mathcal{D}$ or $\mathcal{C} \supseteq \mathcal{D}$ is called a *set constraint* for $\mathcal{C}$.

Further, let set be a set of set constraints for $\mathcal{C}$. Then, a *universal* and an *existential set quantification* is respectively is a formula of the form

$$\forall_{\mathsf{set}}\mathcal{C}.\phi \quad \text{and} \quad \exists_{\mathsf{set}}\mathcal{C}.\phi.$$

The meaning of a set quantification $\forall_{\mathsf{set}}\mathcal{C}.\phi$ (resp. $\exists_{\mathsf{set}}\mathcal{C}.\phi$) is that the variable $\mathcal{C}$ ranges over all subsets of Const_Σ which solve the set constraints in set and, whenever $\mathcal{C}$ occurs in a domain constraint in ϕ, then $\mathcal{C}$ is replaced by the respective subset of Const_Σ. This idea is reflected in the formal definition of interpretations of set quantifications which uses the following definition of solution sets for set constraints:

$$\begin{aligned}\mathsf{Sol}_\Sigma(\mathsf{set}) = \{\mathsf{Const} \subseteq \mathsf{Const}_\Sigma \mid {}& \forall(\mathcal{C} \subseteq \mathcal{D}) \in \mathsf{set}\colon \mathsf{Const} \subseteq \mathcal{D},\\ & \forall(\mathcal{C} \supseteq \mathcal{D}) \in \mathsf{set}\colon \mathsf{Const} \supseteq \mathcal{D}\}.\end{aligned}$$

Definition 2.3.8: Interpretation of Set Quantification

(based on [Van den Broeck, 2013])

Let $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ be a finite signature, let $\phi \in \mathcal{RL}(\Sigma)$ be a formula, let $\mathcal{C}$ be a set variable, and let $I \in \mathsf{Int}(\Sigma)$ be an interpretation. Then, we define

- $I(\forall_{\mathsf{set}}\mathcal{C}.\phi) = \begin{cases} 1 & \text{if } \forall \mathsf{Const} \in \mathsf{Sol}_\Sigma(\mathsf{set})\colon I(\phi\langle\mathcal{C}/\mathsf{Const}\rangle) = 1 \\ 0 & \text{otherwise} \end{cases}$,
- $I(\exists_{\mathsf{set}}\mathcal{C}.\phi) = \begin{cases} 1 & \text{if } \exists \mathsf{Const} \in \mathsf{Sol}_\Sigma(\mathsf{set})\colon I(\phi\langle\mathcal{C}/\mathsf{Const}\rangle) = 1 \\ 0 & \text{otherwise} \end{cases}$,

where $\phi\langle\mathcal{C}/\mathsf{Const}\rangle$ is ϕ in which every occurrence of $\mathcal{C}$ is syntactically replaced by the set of constants Const.

We can express set quantification by conjunctions resp. disjunctions over sets:

- $\exists_{\mathsf{set}}\mathcal{C}.\phi \equiv \bigvee_{\mathsf{Const}\subseteq\mathsf{Const}_\Sigma} \phi\langle\mathcal{C}/\mathsf{Const}\rangle$,
- $\forall_{\mathsf{set}}\mathcal{C}.\phi \equiv \bigwedge_{\mathsf{Const}\subseteq\mathsf{Const}_\Sigma} \phi\langle\mathcal{C}/\mathsf{Const}\rangle$.

Example 2.3.9
Let $\mathsf{Const}_\Sigma = \{a, b\}$. We consider the sentence

$$\phi_{\mathsf{ex}} = \forall_{\{\mathcal{C}\supseteq\{a\}\}}\mathcal{C}.\exists_{\{X\in\mathcal{C}\}}X.A(X)$$

and interpret it by $I \in \mathsf{Int}(\Sigma)$ with $I(A(a)) = 1$ and $I(A(b)) = 0$. We get

$$\begin{aligned} I(\phi_{\mathsf{ex}}) &= I(\forall_{\{\mathcal{C}\supseteq\{a\}\}}\mathcal{C}.\exists_{\{X\in\mathcal{C}\}}X.A(X)) \\ &= I(\exists_{X\in\{a\}}X.A(X) \wedge \exists_{X\in\{a,b\}}X.A(X)) \\ &= I(A(a) \wedge (A(a) \vee A(b))) \\ &= I(A(a)) = 1. \end{aligned}$$

We maintain the notation $\mathcal{RL}(\Sigma)$ for relational logics while also allowing for formulas with domain constraints and set quantification. Note that one has to slightly modify the definition of sentences in this case. In the classical sense, formulas are sentences iff all variables are bounded by a quantification. Here, in addition, we require that all set variables are bounded by a set quantification, too. Formulas which satisfy both requirements are called *sentence*. The set of all these sentences is $\mathcal{RL}^S(\Sigma)$.

2.4 Description Logics

Description Logics ($\mathcal{DL}$) [Baader et al., 2007; Baader and Sattler, 2001; Calvanese et al., 2001] constitute a family of logic-based knowledge representation languages which are especially designed to describe and reason with conceptional knowledge about a specific application domain, also known as *ontologies* [Staab and Studer, 2009]. Common Description Logics are more expressive than propositional logics but less expressive than full first-order logics with the goal to trade off their expressivity against efficient reasoning methods. In particular, reasoning problems for Description Logics are usually decidable. Most notably, Description Logics provide the logical background of the *semantic web* as the *web ontology language* $\mathcal{OWL}$ [Horrocks et al., 2003] is based on Description Logics.

In this section, we give a brief introduction to the basic concepts of Description Logics with a focus on the prototypical Description Logic $\mathcal{ALC}$. We mainly follow the elaborations in [Baader et al., 2008a]. For a more general introduction to Description Logics, we refer to [Baader et al., 2007, 2017, 2008a; Rudolph, 2011].

▶ Description Logic $\mathcal{ALC}$

The Description Logic $\mathcal{ALC}$ was introduced in [Schmidt-Schauß and Smolka, 1991] first. We recall the formal definitions of the syntax and semantics of $\mathcal{ALC}$ here.

The *signature* of $\mathcal{ALC}$ consists of three disjoint sets of *individual names*, *concept names*, and *role names*, in symbols $\mathcal{N}_I$, $\mathcal{N}_C$, and $\mathcal{N}_R$, respectively. As the term suggests, individual names serve as formal representations of individuals (or objects) like constants in relational logics. Concept names stand for basic concepts or properties of individuals, while role names represent binary relationships between them. More complex concepts than concept names can be built in $\mathcal{ALC}$ as follows.

Definition 2.4.1: $\mathcal{ALC}$-Concept *(cf. [Baader et al., 2008a])*

Let $\mathcal{N}_C$ be a set of concept names, and let $\mathcal{N}_R$ be a set of role names. The set of *$\mathcal{ALC}$-concepts* is the smallest set such that $\top$, $\bot$, and every concept name $A \in \mathcal{N}_C$ is an $\mathcal{ALC}$-concept, and, in addition, if C and D are $\mathcal{ALC}$-concepts and $r \in \mathcal{N}_R$ is a role name, then

$$C \sqcap D, \quad C \sqcup D, \quad \neg C, \quad \forall r.C, \quad \exists r.C$$

are $\mathcal{ALC}$-concepts.

The concept constructors $\sqcap$, $\sqcup$, and $\neg$ in Definition 2.4.1 correspond to the conjunction, disjunction, and negation in first-order logics. Quantification in Description Logics is usually restricted in some way, though. In $\mathcal{ALC}$-concepts, quantifiers always occur in combination with role names, for instance. Formally, the semantics of $\mathcal{ALC}$-concepts is given by *interpretations* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consisting of a *domain* $\Delta^\mathcal{I}$ and an *interpretation function* $\cdot^\mathcal{I}$.

Definition 2.4.2: $\mathcal{ALC}$-Semantics *(cf. [Baader et al., 2008a])*

An *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ is a tuple consisting of a non-empty set $\Delta^\mathcal{I}$ (*domain*) and a mapping $\cdot^\mathcal{I}$ (*interpretation function*) which maps every $\mathcal{ALC}$-concept to a subset of $\Delta^\mathcal{I}$ and every role name to a subset of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$ such that for all $\mathcal{ALC}$-concepts C and D and for all role names $r \in \mathcal{N}_R$, it holds that

$$\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}}, \\
\bot^{\mathcal{I}} &= \emptyset, \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}}, \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}}, \\
\neg C^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}, \\
(\exists r.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} \colon (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}, \\
(\forall r.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}} \colon (x,y) \in r^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}.
\end{aligned}$$

In order to distinguish interpretations in Description Logics from interpretations in relational logics, we also use the term *$\mathcal{DL}$-interpretation* instead of *interpretation* for $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$.

In $\mathcal{ALC}$, there are basically two types of statements/sentences, namely *general concept inclusions* and *assertions*. General concept inclusions are used to formalize inclusions between $\mathcal{ALC}$-concepts.

Definition 2.4.3: General Concept Inclusion *(cf. [Baader et al., 2008a])*

Let C and D be $\mathcal{ALC}$-concepts. Then, $C \sqsubseteq D$ is called a *general concept inclusion*. A $\mathcal{DL}$-interpretation $\mathcal{I}$ is a *model* of a general concept inclusion $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds. A finite set of general concept inclusions is called a TBox. A $\mathcal{DL}$-interpretation is a *model* of a TBox $\mathcal{T}$ iff it models all general concept inclusions in $\mathcal{T}$.

The pair of the two general concept inclusions $C \sqsubseteq D$ and $D \sqsubseteq C$ where C and D are $\mathcal{ALC}$-concepts is abbreviated by $C \equiv D$. If A is a concept name and C an $\mathcal{ALC}$-concept, then $A \equiv C$ is called a *definition* of A.

In contrast to general concept inclusions, *assertions* formalize statements about single individuals or about relations between pairs of individuals.

Definition 2.4.4: Assertion *(cf. [Baader et al., 2008a])*

Let C be an $\mathcal{ALC}$-concept, $r \in \mathcal{N}_R$ a role name, and $a, b \in \mathcal{N}_I$ individual names. Then, an *assertion* is an expression either of the form $C(a)$ or $r(a,b)$. A $\mathcal{DL}$-interpretation $\mathcal{I}$ is a *model* of an assertion $C(a)$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$, and it is a model of an assertion $r(a,b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. A finite set of assertions is called an ABox. A $\mathcal{DL}$-interpretation is a *model* of an ABox $\mathcal{A}$ iff it models all assertions in $\mathcal{A}$.

Description Logics are particularly tailored towards representing knowledge. We define an *$\mathcal{ALC}$-knowledge base* as a tuple of a TBox and an ABox.

Definition 2.4.5: $\mathcal{ALC}$-Knowledge Base *(cf. [Baader et al., 2008a])*

A tuple $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consisting of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ is called an $\mathcal{ALC}$-knowledge base. A $\mathcal{DL}$-interpretation $\mathcal{I}$ is a *model* of an $\mathcal{ALC}$-knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ iff $\mathcal{I}$ models both $\mathcal{T}$ and $\mathcal{A}$. If an $\mathcal{ALC}$-knowledge base has a model, then the knowledge base is called *consistent*.

If a $\mathcal{DL}$-interpretation $\mathcal{I}$ is a model of an expression $\mathcal{E}$ where $\mathcal{E}$ is a general concept inclusion, an assertion, an ABox, a TBox, or an $\mathcal{ALC}$-knowledge base, then we indicate this by $\mathcal{I} \models \mathcal{E}$. Typical reasoning tasks in $\mathcal{ALC}$ are deciding whether an $\mathcal{ALC}$-concept is satisfiable or not, as well as answering whether an inference can be drawn from an $\mathcal{ALC}$-knowledge base or not.

Definition 2.4.6: $\mathcal{ALC}$-Reasoning *(cf. [Baader et al., 2008a])*

Let $\mathcal{K}$ be a consistent $\mathcal{ALC}$-knowledge base and let C and D be $\mathcal{ALC}$-concepts. Then, C is called *satisfiable* with respect to $\mathcal{K}$ if there is a model $\mathcal{I}$ of $\mathcal{K}$ with $C^{\mathcal{I}} \neq \emptyset$. The concept D *subsumes* C with respect to $\mathcal{K}$, in symbols $\mathcal{K} \models C \sqsubseteq D$, iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all models $\mathcal{I}$ of $\mathcal{K}$. An individual name $a \in \mathcal{N}_I$ is an instance of C with respect to $\mathcal{K}$, in symbols $\mathcal{K} \models C(a)$, iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$. Finally, a pair of individual names (a, b) is an instance of a role name $r \in \mathcal{N}_R$ with respect to $\mathcal{K}$, written $\mathcal{K} \models r(a, b)$, iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ holds for all models $\mathcal{I}$ of $\mathcal{K}$.

We give an illustrative example.

Example 2.4.7
We consider the $\mathcal{ALC}$-knowledge base $\mathcal{K}_{\mathsf{mth}} = (\mathcal{T}_{\mathsf{mth}}, \mathcal{A}_{\mathsf{mth}})$ with

$$\begin{aligned}
\mathcal{T}_{\mathsf{mth}} = \{ \quad & \mathsf{Parent} && \equiv \mathsf{Person} \sqcap \exists \mathsf{hasChild}.\top, \\
& \mathsf{Grandparent} && \equiv \mathsf{Person} \sqcap \exists \mathsf{hasChild}.\mathsf{Parent}, \\
& \mathsf{Woman} && \sqsubseteq \mathsf{Person} \qquad \}, \\
\mathcal{A}_{\mathsf{mth}} = \{ \quad & \mathsf{Woman}(\mathsf{mary}),\ \mathsf{hasChild}(\mathsf{mary}, \mathsf{paul}) && \qquad \}.
\end{aligned}$$

The knowledge base $\mathcal{K}_{\mathsf{mth}}$ states that parents are defined as persons who have at least one child, grandparents are persons who have at least one child who is a parent themselves, and women are persons. In addition, the ABox of $\mathcal{K}_{\mathsf{mth}}$ tells us that Mary is a woman and has Paul as a child.

For instance, we can infer from $\mathcal{K}_{\mathsf{mth}}$ that Mary is a parent, denoted by

$\mathcal{K}_{\mathsf{mth}} \models \mathsf{Parent}(\mathsf{mary})$, because for every model $\mathcal{I}$ of $\mathcal{K}_{\mathsf{mth}}$ it holds that

$$\begin{aligned} &\mathcal{I} \models (\mathsf{Parent} \equiv \mathsf{Person} \sqcap \exists \mathsf{hasChild}.\top) \\ \Leftrightarrow\ &\mathsf{Parent}^{\mathcal{I}} = (\mathsf{Person} \sqcap \exists \mathsf{hasChild}.\top)^{\mathcal{I}} \\ \Leftrightarrow\ &\mathsf{Parent}^{\mathcal{I}} = \mathsf{Person}^{\mathcal{I}} \cap (\exists \mathsf{hasChild}.\top)^{\mathcal{I}}. \end{aligned}$$

With

$$\mathcal{I} \models \mathsf{Woman}(\mathsf{mary}) \quad \text{i.e.,} \quad \mathsf{mary}^{\mathcal{I}} \in \mathsf{Woman}^{\mathcal{I}},$$

and

$$\mathcal{I} \models (\mathsf{Woman} \sqsubseteq \mathsf{Person}) \quad \text{i.e.,} \quad \mathsf{Woman}^{\mathcal{I}} \subseteq \mathsf{Person}^{\mathcal{I}},$$

it follws that $\mathsf{mary}^{\mathcal{I}} \in \mathsf{Person}^{\mathcal{I}}$ holds. Further,

$$\mathcal{I} \models \mathsf{hasChild}(\mathsf{mary}, \mathsf{paul}) \quad \text{i.e.,} \quad (\mathsf{mary}^{\mathcal{I}}, \mathsf{paul}^{\mathcal{I}}) \in \mathsf{hasChild}^{\mathcal{I}}$$

implies $\mathsf{mary}^{\mathcal{I}} \in (\mathsf{hasChild}.\top)^{\mathcal{I}}$ because

$$\begin{aligned} (\mathsf{mary}^{\mathcal{I}}, \mathsf{paul}^{\mathcal{I}}) &\in \{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} \colon (x, y) \in \mathsf{hasChild}^{\mathcal{I}}\} \\ &= (\mathsf{hasChild}.\top)^{\mathcal{I}}. \end{aligned}$$

Together, we have $\mathsf{mary}^{\mathcal{I}} \in \mathsf{Person}^{\mathcal{I}} \cap (\exists \mathsf{hasChild}.\top)^{\mathcal{I}} = \mathsf{Parent}^{\mathcal{I}}$ and, hence, $\mathcal{I} \models \mathsf{Person}(\mathsf{mary})$. Similarly, we can infer that every grandparent is a parent, $\mathcal{K}_{\mathsf{mth}} \models (\mathsf{Grandparent} \sqsubseteq \mathsf{Parent})$.

In $\mathcal{ALC}$, the reasoning tasks that can be deduced from Definition 2.4.6 can be reduced to checking the consistency of a knowledge base. For instance, $(\mathcal{T}, \mathcal{A}) \models C(a)$ holds iff the knowledge base $(\mathcal{T}, \mathcal{A} \cup \{\neg C(a)\})$ is inconsistent [Baader et al., 2008a].

▶ Extensions and Restrictions of $\mathcal{ALC}$

As indicated in the introduction of this section, a main application of Description Logics is to serve as a formal basis for ontology languages such as OWL. Therefore, many extensions of $\mathcal{ALC}$ have emerged which are especially tailored towards the specific demands of ontologies.

Two basic strategies for extending $\mathcal{ALC}$ are introducing further *concept constructors* and allowing for *role axioms*. In the following, we list some important extensions of $\mathcal{ALC}$ and start with two additional concept constructors.

Nominal Concepts. A *nominal concept* $\{a\}$ where $a \in \mathcal{N}_I$ is a concept which is interpreted by the singleton $\{a^{\mathcal{I}}\}$. Nominal concepts introduce individual names into the TBox. Note that in $\mathcal{ALC}$-knowledge bases, individual names occur in the ABox only.

Number Restrictions. With *number restrictions*, it is possible to constrain the number of relationships a particular type of individuals participates in. Number restrictions are of the form $\geq nr.C$ (*at-least restriction*) or $\leq nr.C$ (*at-most restriction*) where $r \in \mathcal{N}_R$ is a role name, C is an $\mathcal{ALC}$-concept, and n is a natural number. The idea behind the concept $\geq nr.C$ (resp. $\leq nr.C$) is to express that there are at least (at most) n role successors which satisfy the concept C. Number restrictions where $C = \top$, i.e., restrictions of the form $\geq nr.\top$ or $\leq nr.\top$ are called *unqualified*.

In $\mathcal{ALC}$, role names serve as a tool to build complex concepts but do not occur on their own. In some more expressive Description Logics, knowledge bases mention besides the ABox and the TBox also an RBox, though, offering *role axioms* with which relations between roles can be specified. Roles which occur in role axioms can either be role names or *non-simple roles* such as the concatenation of roles $r \circ s$ or inverse roles r^-. We illustrate both role constructors by means of the following example. The role axiom

$$\mathsf{hasChild} \circ \mathsf{hasChild} \sqsubseteq \mathsf{hasGrandparent}^-$$

is the $\mathcal{DL}$-equivalent of the first-order sentence

$$\forall X, Y, Z.(\mathsf{hasChild}(X, Y) \wedge \mathsf{hasChild}(Y, Z) \Rightarrow \mathsf{hasGrandparent}(Z, X))$$

and states that if Y is a child of X and Z is a child of Y, then X is a grandparent of Z. Note the inverted order of the variables X and Z in the term of $\mathsf{hasGrandparent}(Z, X)$ which is due to the inversion of the $\mathcal{DL}$-role $\mathsf{hasGrandparent}$. While the Description Logic $\mathcal{ALC}$ belongs to the two-variable fragment of first-order logic [Lutz et al., 2001], this example shows that Description Logics with role axioms as illustrated above are no longer necessarily in the two-variable fragment of first-order logic. Two-variable logics are important fragments of first-order logic in view of decidability [Grädel et al., 1997].

General *role axioms* are of the form $r_1 \circ \ldots \circ r_n \sqsubseteq s$ where $r_1, \ldots, r_n$ and s are roles or their inverses. Some special instances are:

Role Inclusion. A *role inclusion* $r \sqsubseteq s$ is a general role axiom with $n = 1$ and states that r is a sub-role of s, i.e., $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$.

Transitivity Statement. A *transitivity statement* is of the form $r \circ r \sqsubseteq r$ and states that the role r is transitive. For example,

$$\mathsf{friendOf} \circ \mathsf{friendOf} \sqsubseteq \mathsf{friendOf}$$

corresponds to

$$\forall X, Y, Z.\mathsf{friendOf}(X, Y) \wedge \mathsf{friendOf}(Y, Z) \Rightarrow \mathsf{friendOf}(X, Z)$$

stating that the friendship relation is transitive.

Symmetry Statement. A *symmetry statement* is of the form $r^{-} \sqsubseteq r$ and expresses that r is a symmetric role. For instance,

$$\mathsf{friendOf}^{-} \sqsubseteq \mathsf{friendOf}$$

states that being friends is a symmetric relationship:

$$\forall X, Y.\mathsf{friendOf}(Y, X) \Rightarrow \mathsf{friendOf}(X, Y).$$

In order to find a way through the jungle of Description Logics that has emerged, the following naming scheme has been established (cf. [Schmidt-Schauß and Smolka, 1991; Rudolph, 2011]):

$$((\mathcal{AL}[\mathcal{C}] \mid \mathcal{S})\ [\mathcal{H}] \mid \mathcal{SR})\ [\mathcal{O}]\ [\mathcal{I}]\ [\mathcal{F} \mid \mathcal{N} \mid \mathcal{Q}]$$

where

- $\mathcal{AL}$ stands for a Description Logic which allows for a restricted negation (the so-called *atomic negation* where negations may occur in front of concept names that do not appear on the left-hand side of axioms only), conjunction, universal restriction, and a limited form of existential restriction,
- the additional name constituent $\mathcal{C}$ stands for *complement* and means that in $\mathcal{ALC}$, besides the concept constructors from $\mathcal{AL}$, the negation of complex concepts is allowed which implies the missing concept constructors from Definition 2.4.1 as well,
- $\mathcal{S}$ is an abbreviation for the Description Logic $\mathcal{ALC}$ in which transitivity statements are allowed, in addition,
- $\mathcal{H}$ stands for role inclusions,
- $\mathcal{SR}$ denotes $\mathcal{ALC}$ in which further kinds of role axioms and self concepts are allowed (cf. [Rudolph, 2011] for the details),
- $\mathcal{O}$ indicates support for nominal concepts,
- $\mathcal{I}$ stands for role inverses,

- $\mathcal{F}$ means that functionality statements are allowed (cf. [Rudolph, 2011]),
- $\mathcal{N}$ indicates that unqualified number restrictions are allowed,
- and $\mathcal{Q}$ allows for (qualified) number restrictions.

Besides $\mathcal{ALC}$, further important representatives of Description Logics are $\mathcal{SROIQ}$ [Horrocks et al., 2006] as well as the lightweight Description Logics $\mathcal{EL}$ (cf. [Baader et al., 2005, 2008b]) and the *$\mathcal{DL}$-Lite family* [Calvanese et al., 2005]. The Description Logic $\mathcal{SROIQ}$ "serves as the basis for OWL 2 DL, the most expressive member of the OWL family where inferencing is still decidable" [Rudolph, 2011]. In contrast to the very expressive Description Logic $\mathcal{SROIQ}$, the Description Logic $\mathcal{EL}$ and the $\mathcal{DL}$-Lite provide tractable reasoning [Calvanese et al., 2005; Baader et al., 2008b].

2.5 Comparison of the Logics

We conclude this chapter with a comparison of the introduced logics. All of them are *classical logics* (cf., e.g., [Bergmann et al., 2013; Boolos et al., 2007] and, thus, are capable of representing statements that are true or false. Nevertheless, the logics have their unique characteristics which we want to work out.

First, we show that, in some sense, relational logics are not more expressive than propositional logics and discuss why we consider relational logics instead of propositional logics nevertheless. Afterwards, we compare relational logics to Description Logics and embed the Description Logic $\mathcal{ALC}$ with respect to fixed finite domains into an appropriate relational logic. We conclude this section by noting some limitations of relational logics which are the reasons why relational logics are not used to express knowledge in this thesis directly but are extended to a logic of *relational probabilistic conditionals*.

▶ Propositional Versus Relational Logics

Although propositional logics $\mathcal{L}(\Sigma_P)$ and relational logics $\mathcal{RL}(\Sigma)$ share many commonalities like syntactical elements, in particular the connectives $\neg$, $\wedge$, and $\vee$, and the fact that both kinds of logics satisfy the principle of bivalence, one could think that the possibility to quantify over variables enhances the expressivity of relational logics compared to propositional logics. However, this is not the case provided that Σ_P and Σ are finite and one compares propositional formulas with relational sentences (i.e., relational formulas without free variables). In order to show this, we translate propositional formulas into relational sentences and vice versa.

The easiest way of mimicking a propositional logic by a relational logic is to associate the atomic propositions $\phi \in \Sigma_P$ with nullary predicates $P_\phi/0$, i.e., by introducing one nullary predicate for each atomic proposition. Then, the propositional logic $\mathcal{L}(\Sigma_P)$ can be translated into the relational logic $\mathcal{RL}(\Sigma_{\mathcal{L}(\Sigma_P)})$ with $\Sigma_{\mathcal{L}(\Sigma_P)} = (\emptyset, \{P_\phi/0 \mid \phi \in \Sigma_P\})$. The translation of formulas $\psi \in \mathcal{L}(\Sigma_P)$ into sentences in $\mathcal{RL}^S(\Sigma_{\mathcal{L}(\Sigma_P)})$ is straightforward and can be done by a syntactic replacement of every occurrence of each atomic proposition ϕ that is mentioned in ψ by the corresponding ground atom P_ϕ. In particular, P_ϕ does not mention a variable and, thus, is a sentence. Also the interpretations can be transferred easily: A propositional interpretation assigns to a propositional atom ϕ the value true iff P_ϕ is true in the corresponding relational interpretation. Further, the interpretation of negations $(\neg\phi)$, conjunctions $(\phi \wedge \psi)$, and disjunctions $(\phi \vee \psi)$ is defined the same in propositional and relational logics.

The other way around, sentences from a relational logic $\mathcal{RL}(\Sigma)$ can be expressed by appropriate propositional formulas, too. A way of realizing any sentence $\phi \in \mathcal{RL}^S(\Sigma)$ by a propositional formula is as follows. First, every universal quantification $\forall X.\psi$ in ϕ is equivalently replaced by the conjunction $\bigwedge_{c\in\mathsf{Const}_\Sigma} \psi\langle X/c\rangle$ and every existential quantification $\exists X.\psi$ in ϕ is replaced by the disjunction $\bigvee_{c\in\mathsf{Const}_\Sigma} \psi\langle X/c\rangle$. Because of the finiteness of Const_Σ, the sentence ϕ remains a finite expression. As a result, ϕ is free of variables and every atom which occurs in ϕ is ground. Afterwards, ϕ is translated into a propositional formula by associating every ground atom from $\mathsf{grAtom}(\Sigma)$ with a fresh atomic proposition. That is, $\mathcal{RL}(\Sigma)$ becomes the propositional logic $\mathcal{L}(\mathsf{Ground}(\Sigma))$. Then, the translation of ϕ into a propositional formula works the same as the other way around (see above).

Since every propositional formula can be translated into a relational sentence and vice versa, there is no obvious reason why to prefer relational logics over propositional logics, at least from the viewpoint of expressivity. Nevertheless, there are good arguments in favor of relational logics:

- Relational signatures are especially suitable for knowledge representation:

  The distinction between constants and predicates in signatures of relational logics allows for a clear separation between individuals on the one hand and of properties of and relations between individuals on the other hand. In propositional logics, all these entities would have to be represented on the same syntactical level by atomic propositions. Hence, relational logics better fit to the requirements of knowledge representation in form of *ontologies* [Schmidt-Schauß and Smolka, 1991; Guarino et al., 2009], where individuals, properties, and relations are fundamental concepts to describe the domain of discourse.

- Quantifications convey symmetries and indistinguishabilities:

  Although quantifications in finite relational logics $\mathcal{RL}(\Sigma)$ can be reduced to conjunctions or disjunctions because of the finiteness of the set of constants Const_Σ, quantifications create added value in view of computational aspects as they implicitly express symmetries between and indistinguishabilities of individuals on a syntactical level. This is because unnamed constants, i.e., constants which do not occur in relational expressions explicitly, are interchangeable. These symmetries and indistinguishabilities get lost when the quantifications are resolved. Exploiting symmetries and indistinguishabilities is an essential tool for efficient knowledge compilation [Darwiche and Marquis, 2002] and will play an important role in Chapter 6.

- Domain size as a parameter:

  Recall that we interpret sentences in relational logics $\mathcal{RL}(\Sigma)$ based on a Herbrand semantics [Herbrand, 1930]. This implies that the (syntactical) constants in Const_Σ are in a one-to-one correspondence to the (semantical) objects in the domain of discourse. That is, we apply the *unique name assumption* [Russell and Norvig, 2010], which states that different constants do not refer to the same semantical object, and assume that every domain element is represented by a constant. As a consequence, we may interpret $k = |\mathsf{Const}_\Sigma|$ as the domain size. By treating k as a parameter, we can analyze relational expressions for different domain sizes simultaneously. Also, by varying k, we can investigate the impact of the domain size on the evaluation of the expressions.

- Open formulas extend expressivity:

  In Chapter 3, we will extend relational logics $\mathcal{RL}(\Sigma)$ to logics of qualitative and probabilistic conditionals. These conditionals are built upon relational formulas which do not necessarily have to be sentences. The equiexpressivity which holds between propositional and relational logics is based on a comparison of propositional formulas and relational sentences, though. Relational formulas with free variables bring along an additional degree of freedom in how to interpret them. While probabilistic conditionals defined over a propositional language are canonically interpreted by conditional probabilities, relational probabilistic conditionals require a richer semantics. Here, we will rely on the *aggregating semantics* (cf. Section 3.4) which yields a more elaborate view on probabilistic conditionals than propositional probabilistic conditionals provide.

▶ Relational Logics Versus Description Logics

We identify three major differences between relational logics and the prototypical Description Logic $\mathcal{ALC}$: In contrast to relational logics, $\mathcal{ALC}$ is not subject to the *unique name assumption*, the domain $\Delta^{\mathcal{I}}$ is not necessarily finite, and the possibilities to construct relational expressions in $\mathcal{ALC}$ are limited.

The fact that $\mathcal{ALC}$ is not subject to the *unique name assumption* means that two different individual names $a, b \in \mathcal{N}_I$, $a \neq b$, may be interpreted by the same domain element from $\Delta^{\mathcal{I}}$, i.e., $a^{\mathcal{I}} = b^{\mathcal{I}}$ may hold. This is typical for Description Logics [Baader et al., 2007, 2017] and stands in contrast to relational logics based on an Herbrand semantics—as considered in this thesis—, where no distinction is made between constants and domain elements and, thus, different constants represent different individuals and every domain element is represented by a constant.

In addition, notions such as *satisfiability* and *consistency* can be understood differently. In $\mathcal{ALC}$, satisfiability and consistency are usually not defined with respect to a particular domain $\Delta^{\mathcal{I}}$ as the domain is part of the interpretation $\mathcal{I}$, while we defined Σ-consistency and Σ-satisfiability in relational logics with respect to a prespecified signature Σ. Hence, in $\mathcal{ALC}$ it is sufficient to find a model of an expression with respect to *any* domain $\Delta^{\mathcal{I}}$, in particular of any size $|\Delta^{\mathcal{I}}|$, in order to prove its satisfiability. In the context of probabilistic reasoning, it will be useful to fix a specific domain, though, because interpretations (resp. so-called *possible worlds*) will serve as elementary events of the probability space and, thus, the meaning of the sample space would be ambiguous if the interpretations refer to different domains. Hence, in the probabilistic context, it is quite common to fix a domain and to rely on the unique name assumption, as we have already considered in the definitions of relational logics. In the $\mathcal{DL}$-context, one talks about *fixed (finite) domain reasoning* then (cf. [Gaggl et al., 2016]).

The consideration of infinite domains is an essential component of Description Logics. Handling (countably) infinite domains is challenging in the context of probabilistic reasoning, though, at least in combination with the *principle of maximum entropy* (cf. Section 7.3). Therefore, we will investigate finite domains first and dedicate with Section 7.3 a separate section to the task of incorporating infinite domains into maximum entropy reasoning. Consequently, we will consider from now on $\mathcal{ALC}$ with a *fixed finite domain* (cf. [Gaggl et al., 2016]) until indicated otherwise. We denote this restricted version of $\mathcal{ALC}$ with $\mathcal{ALC}^{\mathsf{ffd}}$.

With respect to the limitations of concept and role constructors in $\mathcal{ALC}$, we can note that $\mathcal{ALC}^{\mathsf{ffd}}$ constitutes a fragment of an appropriate relational logic. When embedding $\mathcal{ALC}^{\mathsf{ffd}}$ into relational logics we can analyze expressions more flexible because we have recourse to the full syntax of the relational logics. This justifies our decision to consider relational logics as background logics for our further elaborations first, and to apply our findings to $\mathcal{ALC}^{\mathsf{ffd}}$ at the end of the thesis. Consequently, we now propose a translation of $\mathcal{ALC}^{\mathsf{ffd}}$-expressions into relational expressions.

For a translation of $\mathcal{ALC}^{\mathsf{ffd}}$ into a relational logic $\mathcal{RL}(\Sigma)$, we understand the individual names in $\mathcal{N}_I$ as constants, the concept names in $\mathcal{N}_C$ as unary predicates, and the role names in $\mathcal{N}_R$ as binary predicates. That is, we consider the relational signature $\Sigma(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R) = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ with

$$\mathsf{Const}_\Sigma = \mathcal{N}_I \quad \text{and} \quad \mathsf{Pred}_\Sigma = \{P_A/1 \mid A \in \mathcal{N}_C\} \cup \{R_r/2 \mid r \in \mathcal{N}_R\}. \tag{2.2}$$

Further, we define a mapping $\gamma_{\mathfrak{C}}$ which maps every $\mathcal{ALC}^{\mathsf{ffd}}$-concept to a formula with one free variable in $\mathcal{RL}(\Sigma)$, and, building on this, a mapping γ which maps every general concept inclusion as well as every assertion from $\mathcal{ALC}^{\mathsf{ffd}}$ to a sentence. Such a translation is, for instance, proposed in a broader setting in [Rudolph, 2011] and adapted to $\mathcal{ALC}^{\mathsf{ffd}}$ in the next definition. A similar translation can also be found in [Baader et al., 2008a].

Definition 2.5.1: Translation from $\mathcal{ALC}^{\mathsf{ffd}}$ to $\mathcal{RL}(\Sigma)$

(based on [Rudolph, 2011])

Let $\mathcal{N}_I$, $\mathcal{N}_C$, and $\mathcal{N}_R$ be the sets of individual, concept, and role names of the Description Logic $\mathcal{ALC}^{\mathsf{ffd}}$. Further, let $\mathfrak{C}$ be the set of $\mathcal{ALC}^{\mathsf{ffd}}$-concepts over $\mathcal{N}_R$ and $\mathcal{N}_C$, let Vars be a set of logical variables, and let $\Sigma(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R)$ be the relational signature induced by $\mathcal{N}_I$, $\mathcal{N}_C$, and $\mathcal{N}_R$. Then, we inductively define a mapping $\gamma_{\mathfrak{C}} \colon \mathfrak{C} \times \mathsf{Vars} \to \mathcal{RL}(\Sigma)$ which translates $\mathcal{ALC}^{\mathsf{ffd}}$-concepts to formulas in $\mathcal{RL}(\Sigma)$ by

$$\begin{aligned}
\gamma_{\mathfrak{C}}(\top, X) &= \top, \\
\gamma_{\mathfrak{C}}(\bot, X) &= \bot, \\
\gamma_{\mathfrak{C}}(A, X) &= P_A(X), \\
\gamma_{\mathfrak{C}}(\neg C, X) &= \neg\gamma_{\mathfrak{C}}(C, X), \\
\gamma_{\mathfrak{C}}(C \sqcap D, X) &= \gamma_{\mathfrak{C}}(C, X) \wedge \gamma_{\mathfrak{C}}(D, X), \\
\gamma_{\mathfrak{C}}(C \sqcup D, X) &= \gamma_{\mathfrak{C}}(C, X) \vee \gamma_{\mathfrak{C}}(D, X), \\
\gamma_{\mathfrak{C}}(\exists r.C, X) &= \exists Y.(R_r(X, Y) \wedge \gamma_{\mathfrak{C}}(C, Y)), \qquad & Y \neq X, \\
\gamma_{\mathfrak{C}}(\forall r.C, X) &= \forall Y.(R_r(X, Y) \Rightarrow \gamma_{\mathfrak{C}}(C, Y)), \qquad & Y \neq X,
\end{aligned}$$

where $A \in \mathcal{N}_C$ is a concept name, $C, D \in \mathfrak{C}$ are $\mathcal{ALC}^{\mathsf{ffd}}$-concepts, $r \in \mathcal{N}_R$ is a role name, and $X, Y \in \mathsf{Vars}$ are variables. Based on this mapping, we further define the $\mathcal{ALC}^{\mathsf{ffd}}$*-translation* $\gamma \colon \mathfrak{A} \to \mathcal{RL}^S(\Sigma)$, where $\mathfrak{A}$ is the set of all general concept inclusions and assertions in $\mathcal{ALC}^{\mathsf{ffd}}$, by

$$\gamma(C \sqsubseteq D) = \forall X.(\gamma_{\mathfrak{C}}(C, X) \Rightarrow \gamma_{\mathfrak{C}}(D, X))$$

$$\begin{aligned}\gamma(C(a)) &= \gamma_{\mathfrak{C}}(C, X)\langle X/a\rangle,\\ \gamma(r(a,b)) &= R_r(a,b).\end{aligned}$$

The translation γ can be extended to a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ by

$$\begin{aligned}\gamma(\mathcal{T}) &= \bigwedge_{(C\sqsubseteq D)\in\mathcal{T}} \gamma(C \sqsubseteq D),\\ \gamma(\mathcal{A}) &= \bigwedge_{C(a)\in\mathcal{A}} \gamma(C(a)) \wedge \bigwedge_{r(a,b)\in\mathcal{A}} R_r(a,b).\end{aligned}$$

We give an example.

Example 2.5.2
We recall the $\mathcal{ALC}$-knowledge base $\mathcal{K}_{\mathsf{mth}} = (\mathcal{T}_{\mathsf{mth}}, \mathcal{A}_{\mathsf{mth}})$ from Example 2.4.7 which is given by

$$\begin{aligned}\mathcal{T}_{\mathsf{mth}} = \{\ & \mathsf{Parent} && \equiv \mathsf{Person} \sqcap \exists\mathsf{hasChild}.\top,\\ & \mathsf{Grandparent} && \equiv \mathsf{Person} \sqcap \exists\mathsf{hasChild}.\mathsf{Parent},\\ & \mathsf{Woman} && \sqsubseteq \mathsf{Person} \quad \},\\ \mathcal{A}_{\mathsf{mth}} = \{\ & \mathsf{Woman}(\mathsf{mary}),\ \mathsf{hasChild}(\mathsf{mary}, \mathsf{paul}) && \quad \}.\end{aligned}$$

Then,

$$\begin{aligned}&\gamma(\mathcal{A}_{\mathsf{mth}})\\ &= \gamma(\mathsf{Woman}(\mathsf{mary})) \wedge \gamma(\mathsf{hasChild}(\mathsf{mary}, \mathsf{paul}))\\ &= \gamma_{\mathfrak{C}}(\mathsf{Woman}, X)\langle X, \mathsf{mary}\rangle \wedge R_{\mathsf{hasChild}}(\mathsf{mary}, \mathsf{paul})\\ &= P_{\mathsf{Woman}}(X)\langle X, \mathsf{mary}\rangle \wedge R_{\mathsf{hasChild}}(\mathsf{mary}, \mathsf{paul})\\ &= P_{\mathsf{Woman}}(\mathsf{mary}) \wedge R_{\mathsf{hasChild}}(\mathsf{mary}, \mathsf{paul}),\end{aligned}$$

$$\begin{aligned}&\gamma(\mathcal{T}_{\mathsf{mth}})\\ &= \gamma(\mathsf{Parent} \sqsubseteq \mathsf{Person} \sqcap \exists\mathsf{hasChild}.T)\\ &\quad \wedge \gamma(\mathsf{Person} \sqcap \exists\mathsf{hasChild}.T \sqsubseteq \mathsf{Parent})\\ &\quad \wedge \gamma(\mathsf{Grandparent} \sqsubseteq \mathsf{Person} \sqcap \exists\mathsf{hasChild}.\mathsf{Parent})\\ &\quad \wedge \gamma(\mathsf{Person} \sqcap \exists\mathsf{hasChild}.\mathsf{Parent} \sqsubseteq \mathsf{Grandparent})\\ &\quad \wedge \gamma(\mathsf{Woman} \sqsubseteq \mathsf{Person})\\ &= \forall X.(\gamma_{\mathfrak{C}}(\mathsf{Parent}, X) \Rightarrow \gamma_{\mathfrak{C}}(\mathsf{Person} \sqcap \exists\mathsf{hasChild}.T, X))\\ &\quad \wedge \forall X.(\gamma_{\mathfrak{C}}(\mathsf{Person} \sqcap \exists\mathsf{hasChild}.T, X) \Rightarrow \gamma_{\mathfrak{C}}(\mathsf{Parent}, X))\\ &\quad \wedge \forall X.(\gamma_{\mathfrak{C}}(\mathsf{Grandparent}, X) \Rightarrow \gamma_{\mathfrak{C}}(\mathsf{Person} \sqcap \exists\mathsf{hasChild}.\mathsf{Parent}, X))\end{aligned}$$

$$
\begin{aligned}
&\quad \wedge \forall X.(\gamma_{\mathfrak{C}}(\mathsf{Person} \sqcap \exists \mathsf{hasChild}.\mathsf{Parent}, X) \Rightarrow \gamma_{\mathfrak{C}}(\mathsf{Grandparent}, X))\\
&\quad \wedge \forall X.(\gamma_{\mathfrak{C}}(\mathsf{Woman}, X) \Rightarrow \gamma_{\mathfrak{C}}(\mathsf{Person}, X))\\
&= \forall X.(P_{\mathsf{Parent}}(X) \Rightarrow (P_{\mathsf{Person}}(X) \wedge \exists Y.(R_{\mathsf{hasChild}}(X,Y)))\\
&\quad \wedge \forall X.((P_{\mathsf{Person}}(X) \wedge \exists Y.(R_{\mathsf{hasChild}}(X,Y)) \Rightarrow P_{\mathsf{Parent}}(X))\\
&\quad \wedge \forall X.(P_{\mathsf{Grandparent}}(X) \Rightarrow (P_{\mathsf{Person}}(X) \wedge \exists Y.(R_{\mathsf{hasChild}}(X,Y) \wedge P_{\mathsf{Parent}}(Y)))\\
&\quad \wedge \forall X.((P_{\mathsf{Person}}(X) \wedge \exists Y.(R_{\mathsf{hasChild}}(X,Y) \wedge P_{\mathsf{Parent}}(Y)) \Rightarrow P_{\mathsf{Grandparent}}(X))\\
&\quad \wedge \forall X.(P_{\mathsf{Woman}}(X) \Rightarrow P_{\mathsf{Person}}(X)).
\end{aligned}
$$

The translation γ from Definition 2.5.1 proves that $\mathcal{ALC}^{\mathsf{ffd}}$ is contained in the two-variable fragment of relational logics as well as a *guarded fragment* in the sense of [Grädel, 1998] because quantifications occur in $\mathcal{ALC}^{\mathsf{ffd}}$ in restricted settings only. It is a straightforward result that an $\mathcal{ALC}^{\mathsf{ffd}}$-knowledge base $(\mathcal{T}, \mathcal{A})$ is consistent (wrt. the fixed domain) iff the relational sentence $\gamma(\mathcal{T}) \wedge \gamma(\mathcal{A})$ is consistent. Further, when $\mathcal{E}$ is a general concept inclusion or an assertion, then $(\mathcal{T}, \mathcal{A}) \models \mathcal{E}$ holds iff $\gamma(\mathcal{T}) \wedge \gamma(\mathcal{A}) \Rightarrow \gamma(\mathcal{E})$ is true in any interpretation defined over the fixed domain (cf. [Baader et al., 2008a]).

▶ Limitations of Relational Logics

We now address some limitations of relational logics $\mathcal{RL}(\Sigma)$. To some extent, we will overcome these limitations in course of the thesis by proper extensions of $\mathcal{RL}(\Sigma)$, especially by introducing relational probabilistic conditionals in Chapter 3.

- Limitations caused by the finite domain:

  Because the signature Σ of relational logics $\mathcal{RL}(\Sigma)$ is finite, especially the number of constants $|\mathsf{Const}_\Sigma|$, it is impossible to differentiate between infinitely many individuals or objects within $\mathcal{RL}(\Sigma)$. This might be tolerable because the number of individuals or objects in the domain of discourse is usually finite (maybe very large and often not known but finite) but, and this is more important, it is also not possible to assign infinitely many states to attributes of entities in $\mathcal{RL}(\Sigma)$. For example, the body mass index (BMI) of a person can take infinitely many values in general as it is technically a positive real number. However, it is impossible to formulate an atom $\mathsf{BMI}(X, Y)$ in $\mathcal{RL}(\Sigma)$ stating that the BMI of a person X is Y where Y ranges over all positive real numbers because the range of Y must be finite by definition of $\mathcal{RL}(\Sigma)$. We will address this issue in Section 7.3. Basically, our approach is based on the idea that even if the BMI can take infinitely many values, as soon as we formulate knowledge about the BMI, we typically group these values into finitely many different classes. For example, in order to detect adiposity, it is relevant for the doctor if the BMI is either less than or equal to 30 or greater than 30. Hence, we can partition all possible values of the BMI into two classes (less than 30 or not) which reduces reasoning to the finite case. Technically, we will realize this idea by utilizing the concept of *satisfiability modulo theories* (cf. [Barrett et al., 2021]).

- Limitations of $\mathcal{RL}(\Sigma)$ as being a two-valued logic:

  In the real world, statements (beliefs) typically do not hold for sure. Thus, it is desirable to be able to formulate statements that hold in most cases, with some kind of uncertainty, under some kind of plausibility assumptions, etc. when applying formal reasoning approaches to practical problems. However, relational logics $\mathcal{RL}(\Sigma)$ satisfy the *principle of bivalence* (cf. e.g., [Beziau, 2003]), i.e., all sentences are mapped to the truth values 1 (*true*) or 0 (*false*) but no truth values in-between are allowed. In addition, the truth values of sentences in $\mathcal{RL}(\Sigma)$ are uniquely defined by the truth values of their subsentences (*principle of extensionality*, cf. [Kunen, 1980]). For these reasons, relational logics are *classical logics* and not suited for capturing the concept of *plausibility* or for representing the *likelihood* of the correctness of a statement. Hence, it is not possible to formalize *uncertain* or *vague knowledge* as well as *preferences* with $\mathcal{RL}(\Sigma)$ (see [Gabbay, 1994] for a comparison of classical and non-classical logics). To overcome this kind of limitation, we extend relational logics to *relational probabilistic conditional logics* $\mathcal{RPCL}(\Sigma)$ in Chapter 3.

- Limitations on the inference formalism:

  The entailment relation $\models_\Sigma$ of relational logics $\mathcal{RL}(\Sigma)$ is *monotonic*, i.e., for sets of sentences $\mathcal{A}_1, \mathcal{A}_2 \subseteq \mathcal{RL}^S(\Sigma)$ and a sentence $B \in \mathcal{RL}^S(\Sigma)$, it holds that if $\mathcal{A}_1 \models_\Sigma B$ and $\mathcal{A}_1 \subseteq \mathcal{A}_2$, then $\mathcal{A}_2 \models_\Sigma B$ holds, too. As a consequence, it is not possible to adequately *revise* knowledge but only on a meta-level.

Proposition 2.5.3: Monotony of Classical Entailment

Let $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ be a finite signature, let $\mathcal{A}_1, \mathcal{A}_2 \subseteq \mathcal{RL}^S(\Sigma)$ be finite sets of sentences and let $B \in \mathcal{RL}^S(\Sigma)$ be a sentence. Then,

$$\mathcal{A}_1 \models_\Sigma B \quad \text{and} \quad \mathcal{A}_1 \subseteq \mathcal{A}_2 \quad \text{imply} \quad \mathcal{A}_2 \models_\Sigma B.$$

Proof. From $\mathcal{A}_1 \subseteq \mathcal{A}_2$ it follows that $\mathsf{Mod}_\Sigma(\mathcal{A}_2) \subseteq \mathsf{Mod}_\Sigma(\mathcal{A}_1)$ holds. Let $I \in \mathsf{Mod}_\Sigma(\mathcal{A}_2)$ be a model of $\mathcal{A}_2$. Then, $I \in \mathsf{Mod}_\Sigma(\mathcal{A}_1)$ holds and with $\mathcal{A}_1 \models_\Sigma B$ it follows that I is a model of B, too. □

Probabilistic inference formalisms as discussed in Sections 4.1 and 4.2 will be *nonmonotonic* and able to capture defeasibility instead.

3

Probabilistic Conditionals and the Aggregating Semantics

In this chapter we introduce *probabilistic conditionals* and the *aggregating semantics* [Kern-Isberner and Thimm, 2010]. We begin in Section 3.1 with extending the relational logics $\mathcal{RL}(\Sigma)$ by a conditional operator that allows us to formalize uncertain, defeasible knowledge like "birds are *usually* able to fly," which usually holds but leaves room for exceptions—here the non-flying penguins, for instance. In terms of relational conditionals, this statement could be formalized as $(\mathsf{Flies}(X)|\mathsf{Bird}(X))$ where X is a variable that can be instantiated by a specific individual. The following Section 3.2 is dedicated to *conditional structures.* Relational conditionals $(B|A)$ without free variables can either be verified (A and B hold), falsified (A holds but B is false), or not applicable (B is false). In case of conditionals with free variables, this three-valued evaluation can be carried out for every single instance. Conditional structures constitute an algebraic representation of the overall evaluation of conditionals. Afterwards, in Section 3.3, conditionals are extended by probabilities in order to quantify their uncertainty. For conditionals without free variables, the common semantics via conditional probabilities is sufficient. So-called *open conditionals* with free variables require a richer semantics, though. In Section 3.4, we discuss possible semantics of open probabilistic conditionals, in particular the aggregating semantics. The aggregating semantics is a semantics of open conditionals which combines statistical and subjective aspects of probabilities.

3.1 Relational Conditionals

Conditionals [Adams, 1975; Nute, 1980] are formal representations of qualitative, uncertain statements where uncertainty refers to defeasibility. Therewith, conditionals constitute a well-founded and prominent way of entering the wide field of *nonmonotonic reasoning* (cf., e.g., [Halpern, 2005; Brewka, 1991; Makinson, 2005]). Defeasibility is inevitable when reasoning in changing environments [Alchourrón et al., 1985] but also in (static) human knowledge representation as it is suited to dissolve apparent contradictions, and it is present in nearly every application domain of formal reasoning, in particular in such sensitive disciplines as medicine, accounting, and law. Relational conditionals combine the defeasibility of conditionals with relational background logics.

In this section we give an intuition on how conditionals differ from material implication in classical logics first. Then, we proceed with a formal definition of (qualitative) relational conditionals. Because we will extend conditionals by probabilities in the next sections, we discuss only those semantical aspects of qualitative conditionals that will be helpful in the probabilistic case as well. More precisely, we focus on the notions of *possible worlds* and *counting functions* which will play an important role in the probabilistic setting, too. For a more in-depth discussion of the semantics of qualitative relational conditionals, we refer to [Kern-Isberner and Thimm, 2012].

▶ Conditionals and Defeasibility

A *conditional* $(\psi|\phi)$, sometimes denoted by $\phi \supset \psi$, is a formal representation of the defeasible statement "If ϕ holds, then *usually* ψ follows." In the narrower sense, a conditional $(\psi|\phi)$ expresses a directed dependency between ϕ (the *premise* of the conditional) and ψ (the *conclusion*) with the vision that ϕ is a (or *the*) reason for ψ while leaving room for uncertainty [Cummins et al., 1991]. More generally, $(\psi|\phi)$ states that in the context of ϕ it is more *plausible* to assume ψ than $\neg\psi$.

The defeasibility of conditionals distinguishes conditionals from expressions in classical logics, in particular from material implication, and clearly shows itself in the three-valued evaluation of conditionals (cf. [de Finetti, 2017]): Let $(\psi|\phi)$ be a conditional where ϕ and ψ are sentences from $\mathcal{RL}^S(\Sigma)$. Then, besides the *verification* of $(\psi|\phi)$, i.e., the case where ϕ and ψ are true, and its *falsification* where ϕ is true but ψ is false, the conditional $(\psi|\phi)$ can also be *not applicable*, namely when the premise ϕ is *false*. This is in contrast to the material implication $\phi \Rightarrow \psi$ which is *true* if ϕ is *false*. Consequently, the conditional $(\psi|\phi)$ may take on the truth values 1 and 0, but also the additional value u (*unknown* or *not applicable*) if ϕ is *false*. This constitution of conditionals allows us to handle exceptions: "If ϕ holds, then usually ψ follows" particularly means that there might be a case in which ψ does not follow from ϕ, i.e., there might be an additional circumstance ρ besides ϕ which causes that ψ does not hold. For example, the conditional $(\mathsf{Ill}(X)|\mathsf{InfectedBy}(X,\mathsf{flu}))$ may state that a person X usually becomes ill when she is infected by the flu. An exceptional circumstance which causes that this conclusion does not hold could be the fact that the person X is vaccinated, has taken high doses of vitamins, or has a generally well developed immune system. Such exceptions need not to be mentioned explicitly in a conditional but are considered implicitly. Although, they can be made explicit by the use of an additional conditional. In our example, for instance, such an additional conditional could be $(\neg\mathsf{Ill}(X)|\mathsf{InfectedBy}(X,\mathsf{flu}) \wedge \mathsf{VaccinatedAgainst}(X,\mathsf{flu}))$.

Note that the conditional $(\mathsf{Ill}(X)|\mathsf{InfectedBy}(X,\mathsf{flu}))$ mentions a free variable and, hence, requires a more sophisticated semantics than the three-valued evaluation of conditionals $(\psi|\phi)$ where ϕ and ψ are sentences. The possibility to handle conditionals with free variables is a strong reason to consider a relational background language instead of a propositional language. The fact that propositional formulas and relational sentences have the same expressivity (cf. Section 2.5) does not necessarily carry over to conditional statements. Actually, this depends on the semantics of conditionals with free variables.

After this brief sketch of the essence of conditionals, we now give a formal introduction into the concept of conditionals as used in this thesis.

► Syntax of Relational Conditionals

In this thesis, conditionals are built upon formulas from a relational background logic $\mathcal{RL}(\Sigma)$. In addition, we make use of domain constraints in order to restrict the scope of conditionals. The domain constraints are in analogy to the domain constraints of quantifiers as introduced in Section 2.3. Recall that domain constraints are expressions of the form $X \neq c$, $X \neq Y$, $X \in \mathcal{C}$, or $X \in \mathsf{Const}_\Sigma \setminus \mathcal{C}$ where X and Y are logical variables, $c \in \mathsf{Const}_\Sigma$ is a constant, and where $\mathcal{C} \subseteq \mathsf{Const}_\Sigma$ is a set of constants. The formal definition of conditionals is as follows.

Definition 3.1.1: (Qualitative Relational) Conditional

Let Σ be a finite signature, let $\phi, \psi \in \mathcal{RL}(\Sigma)$ be formulas, and let CS be a set of domain constraints for the variables in $\mathsf{FreeVar}(\phi) \cup \mathsf{FreeVar}(\psi)$, i.e., for the variables which are free in ϕ or ψ. Then, $\delta = (\psi|\phi)_{\mathsf{CS}}$ is called a *(qualitative relational) conditional*. If $\mathsf{CS} = \emptyset$, then we write $(\psi|\phi)$ instead of $(\psi|\phi)_{\mathsf{CS}}$ for short. We denote the set of all conditionals defined over the signature Σ with $\mathcal{RCL}(\Sigma)$.

While domain constraints in quantifications $\forall_{\mathsf{CS}} X.\phi$ or $\exists_{\mathsf{CS}} X.\phi$ are used to restrict the scope of the free variables in ϕ, the domain constraints of conditionals $(\psi|\phi)_{\mathsf{CS}}$

are used to restrict the scope of the free variables in both ϕ and ψ. Free variables which occur in ϕ as well as in ψ relate the premise and the conclusion of $\delta = (\psi|\phi)_{\mathsf{CS}}$ in that they are placeholders for the same (set of) constants. That is, in the example $(\mathsf{Ill}(X)|\mathsf{InfectedBy}(X, \mathsf{flu}))$, the person X which has the flu is the same which possibly gets ill.

A conditional $(\psi|\phi)_{\mathsf{CS}}$ is *closed* iff both ϕ and ψ are sentences, i.e., iff $\phi, \psi \in \mathcal{RL}^S(\Sigma)$. Otherwise, the conditional $(\psi|\phi)_{\mathsf{CS}}$ is *open*. The constraint set CS of closed conditionals $\delta = (\psi|\phi)_{\mathsf{CS}}$ is empty because $\mathsf{FreeVar}(\delta) = \emptyset$ holds in this case.

Example 3.1.2
We formalize the observation of the sociologist from Example 2.2.2(c) that friends of smokers usually smoke, too. For this, we consider the signature Σ_{smk} with the set of predicates

$$\mathsf{Pred}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{Friends}/2,\ \mathsf{Smokes}/1\}$$

and define the conditional

$$\delta_{\mathsf{smk}} = (\mathsf{Smokes}(X)|\mathsf{Friends}(X, Y) \wedge \mathsf{Smokes}(Y))_{\{X \neq Y\}}.$$

The informal meaning of this conditional is: “If two persons are friends and one of them smokes, then usually the other person smokes, too.” It is important to note that δ_{smk} is a *defeasible* statement. That is, it might be the case that there is a friend of a smoker who does not smoke. But this must be an exception. A formal semantics of conditionals has to define what *usually* and *exception* means then. The conditional δ_{smk} is open because $\mathsf{FreeVar}(\delta_{\mathsf{smk}}) = \{X, Y\} \neq \emptyset$.

Like open formulas, open conditionals from $\mathcal{RCL}(\Sigma)$ can be instantiated. A proper *instance* of an open conditional $\delta = (\psi|\phi)_{\mathsf{CS}}$ is observed by substituting each (free) variable from $\mathsf{FreeVar}(\delta) := \mathsf{FreeVar}(\phi) \cup \mathsf{FreeVar}(\psi)$ with a constant such that both

- free variables that are mentioned in both the premise ϕ and the conclusion ψ of δ, i.e., the variables in $\mathsf{FreeVar}(\phi) \cap \mathsf{FreeVar}(\psi)$, are substituted with the same constant in ϕ as in ψ,
- the domain constraints in CS are satisfied.

The set of all proper instances of a conditional $\delta \in \mathcal{RCL}(\Sigma)$ with respect to the signature Σ is denoted by $\mathsf{Inst}_{\Sigma}(\delta)$. As in the case of open formulas, the set of instances $\mathsf{Inst}_{\Sigma}(\delta)$ depends on Const_{Σ} instead of the whole signature Σ only but we use the subscript Σ for convenience. Instances of conditionals are always closed conditionals.

Example 3.1.3

Again, we consider the signature $\Sigma_{\mathsf{smk}} = (\mathsf{Const}_{\Sigma_{\mathsf{smk}}}, \mathsf{Pred}_{\Sigma_{\mathsf{smk}}})$ from Example 2.2.2(c) with

$$\mathsf{Const}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{peter},\ \mathsf{paul},\ \mathsf{mary}\} \quad \text{and} \quad \mathsf{Pred}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{Friends}/2,\ \mathsf{Smokes}/1\}.$$

The set of the proper instances of the open conditional

$$\delta_{\mathsf{smk}} = (\mathsf{Smokes}(X)|\mathsf{Friends}(X,Y) \wedge \mathsf{Smokes}(Y))_{\{X \neq Y\}}$$

is given by

$$\begin{aligned}
\mathsf{Inst}_{\Sigma_{\mathsf{smk}}}(\delta_{\mathsf{smk}}) = \{&(\mathsf{Smokes}(\mathsf{peter})|\mathsf{Friends}(\mathsf{peter},\mathsf{paul}) \wedge \mathsf{Smokes}(\mathsf{paul})),\\
&(\mathsf{Smokes}(\mathsf{peter})|\mathsf{Friends}(\mathsf{peter},\mathsf{mary}) \wedge \mathsf{Smokes}(\mathsf{mary})),\\
&(\mathsf{Smokes}(\mathsf{paul})|\mathsf{Friends}(\mathsf{paul},\mathsf{peter}) \wedge \mathsf{Smokes}(\mathsf{peter})),\\
&(\mathsf{Smokes}(\mathsf{paul})|\mathsf{Friends}(\mathsf{paul},\mathsf{mary}) \wedge \mathsf{Smokes}(\mathsf{mary})),\\
&(\mathsf{Smokes}(\mathsf{mary})|\mathsf{Friends}(\mathsf{mary},\mathsf{peter}) \wedge \mathsf{Smokes}(\mathsf{peter})),\\
&(\mathsf{Smokes}(\mathsf{mary})|\mathsf{Friends}(\mathsf{mary},\mathsf{paul}) \wedge \mathsf{Smokes}(\mathsf{paul}))\}.
\end{aligned}$$

We now discuss the formal semantics of conditionals which is based on *possible worlds*.

▶ Possible Worlds Semantics

The semantics of conditionals is based on the notion of *possible worlds* [Lewis, 2013; Stalnaker, 1976]. The idea behind possible worlds is to represent all *conceivable* states of the real world in terms of a formalized logical language. Hence, possible worlds are abstract entities which determine what is true and what is false in the particular view on the world. Given some conditionals which formalize knowledge about the real world, one major task of uncertain reasoning is to identify the possible world which describes the real world best.

In this thesis, we associate possible worlds with the truth assignments to the ground atoms in $\mathsf{grAtom}(\Sigma)$ which underlie the interpretations in $\mathsf{Int}(\Sigma)$ (cf. Definition 2.2.7) and represent the possible world ω_θ which refers to the truth assignment $\theta \in \Theta(\Sigma)$ as a complete conjunction of those ground literals which are interpreted to 1 (*true*) by θ. Thus, *possible worlds* are logical representations of the maximal consistent subsets of $\mathsf{grLit}(\Sigma)$.

Definition 3.1.4: Possible World *(cf. [Kern-Isberner and Thimm, 2012])*

Let Σ be a finite signature, and let $\theta \in \Theta(\Sigma)$ be a truth assignment to the ground atoms in $\mathsf{grAtom}(\Sigma)$. Then, the possible world with respect to θ is

$$\omega_\theta = \bigwedge_{l \in \mathsf{grLit}(\Sigma)\colon I_\theta(l)=1} l = \bigwedge_{g \in \mathsf{grAtom}(\Sigma)\colon \theta(g)=1} g \wedge \bigwedge_{g \in \mathsf{grAtom}(\Sigma)\colon \theta(g)=0} \bar{g}.$$

The set of all possible worlds over Σ is denoted with $\Omega(\Sigma)$.

Because possible worlds are represented as conjunctions of ground literals, technically they are sentences, i.e., $\Omega(\Sigma) \subseteq \mathcal{RL}^S(\Sigma)$. This makes it easier to relate possible worlds as we can formulate conjunctions or disjunctions of them.

Example 3.1.5

We consider the signature $\Sigma_{\mathsf{bfp}} = (\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}, \mathsf{Pred}_{\Sigma_{\mathsf{bfp}}})$ (cf. Example 2.2.2(b)) with

$$\mathsf{Const}_{\Sigma_{\mathsf{bfp}}} = \{\mathsf{tweety},\ \mathsf{sally}\} \quad \text{and} \quad \mathsf{Pred}_{\Sigma_{\mathsf{bfp}}} = \{\mathsf{Bird}/1,\ \mathsf{Flies}/1,\ \mathsf{Penguin}/1\}.$$

Then,

$$\begin{aligned}\mathsf{grLit}(\Sigma_{\mathsf{bfp}}) = \{&\mathsf{Bird}(\mathsf{tweety}), \mathsf{Bird}(\mathsf{sally}), \mathsf{Flies}(\mathsf{tweety}), \mathsf{Flies}(\mathsf{sally}),\\ &\mathsf{Penguin}(\mathsf{tweety}), \mathsf{Penguin}(\mathsf{sally}), \neg\mathsf{Bird}(\mathsf{tweety}), \neg\mathsf{Bird}(\mathsf{sally}),\\ &\neg\mathsf{Flies}(\mathsf{tweety}), \neg\mathsf{Flies}(\mathsf{sally}), \neg\mathsf{Penguin}(\mathsf{tweety}),\\ &\neg\mathsf{Penguin}(\mathsf{sally})\}.\end{aligned}$$

An example of a possible world in $\Omega(\Sigma_{\mathsf{bfp}})$ is

$$\begin{aligned}\omega = {}&\mathsf{Bird}(\mathsf{tweety}) \wedge \neg\mathsf{Flies}(\mathsf{tweety}) \wedge \mathsf{Penguin}(\mathsf{tweety})\\ &\wedge \mathsf{Bird}(\mathsf{sally}) \wedge \mathsf{Flies}(\mathsf{sally}) \wedge \neg\mathsf{Penguin}(\mathsf{sally}).\end{aligned}$$

In this possible world ω, Tweety is a penguin bird which is not able to fly while Sally is a flying bird which is not a penguin.

Via the truth assignments in $\Theta(\Sigma)$, possible worlds are in a one-to-one correspondence with interpretations. For all possible worlds $\omega \in \Omega(\Sigma)$ and interpretations $I \in \mathsf{Int}(\Sigma)$, we have $\omega \sim I$ iff $\omega = \omega_\theta$ and $I = I_\theta$ for some $\theta \in \Theta(\Sigma)$. In particular, we have

$$|\Omega(\Sigma)| = 2^{|\mathsf{Ground}(\Sigma)|} = 2^{\sum_{P \in \mathsf{Pred}_\Sigma} |\mathsf{Const}_\Sigma|^{\mathsf{arity}(P)}} = |\mathsf{Int}(\Sigma)|. \tag{3.1}$$

As a consequence, the number of possible worlds in $\Omega(\Sigma)$ is exponential in the number of constants in Const_Σ. This makes naïve reasoning with respect to formalisms based on possible worlds semantics computationally expensive, in particular when $|\mathsf{Const}_\Sigma|$ is large. Controlling the *domain size* $|\mathsf{Const}_\Sigma|$ will be *the* crucial task in this thesis, especially in the context of *lifted inference* (cf. Chapter 5). Therefore, we allocate $k = |\mathsf{Const}_\Sigma|$ throughout the thesis.

Sometimes we will treat k as a parameter as well. This means that we consider signatures Σ with a fixed set of predicates Pred_Σ and a set of constants Const_Σ of arbitrary but finite size.

▶ Marginalization of Possible Worlds

For some reasoning tasks such as *marginalization* it is useful to restrict possible worlds to a subsignature of Σ, i.e., to a signature Σ' with $\Sigma' \sqsubseteq \Sigma$, or more generally to a subset of ground atoms $\mathcal{G} \subseteq \mathsf{grAtom}(\Sigma)$. Note that not all sets of ground atoms $\mathcal{G}$ are induced by a signature Σ, i.e., of the form $\mathsf{grAtom}(\Sigma')$ where Σ' is a signature (cf. Example 3.1.8). The restriction of possible worlds to a subset of ground atoms $\mathcal{G} \subseteq \mathsf{grAtom}(\Sigma)$ is especially useful if the truth assignment on $\mathsf{grAtom}(\Sigma) \setminus \mathcal{G}$ is irrelevant or if we want to *localize* resp. *focus* reasoning on $\mathcal{G}$. For example, we could aim at decomposing a sentence into a set of subsentences which can be interpreted on small subsignatures of Σ such that the restricted interpretations of the subsentences can be assembled to a proper interpretation of the whole sentence.

Definition 3.1.6: Partial Possible World *(cf. [Wilhelm et al., 2017b])*

Let Σ be a finite signature, and let $\mathcal{G} \subseteq \mathsf{grAtom}(\Sigma)$ be a set of ground atoms. We denote with $\Theta(\mathcal{G})$ the set of the truth assignments $\theta_\mathcal{G} \colon \mathcal{G} \to \{0,1\}$ which assign a truth value to the ground atoms in $\mathcal{G}$. For $\theta_\mathcal{G} \in \Theta(\mathcal{G})$, the *partial possible world* with respect to $\theta_\mathcal{G}$ is defined by

$$\omega_{\theta_\mathcal{G}} = \bigwedge_{g \in \mathcal{G} \colon \theta_\mathcal{G}(g)=1} g \wedge \bigwedge_{g \in \mathcal{G} \colon \theta_\mathcal{G}(g)=0} \bar{g}.$$

The set of all partial possible worlds with respect to $\mathcal{G}$ is denoted with $\Omega(\mathcal{G})$. Note that we also use the notation $\Omega(\Sigma')$ instead of $\Omega(\mathcal{G})$ if $\mathcal{G}$ is of the form $\mathcal{G} = \mathsf{grAtom}(\Sigma')$ for a subsignature $\Sigma' \sqsubseteq \Sigma$.

Partial possible worlds can be obtained from possible worlds by *marginalization*. The marginalization of possible worlds $\omega \in \Omega(\Sigma)$ means the projection of ω onto a subset of $\mathsf{grAtom}(\Sigma)$.

Definition 3.1.7: Marginalized Possible World

(cf. [Wilhelm et al., 2017b])

Let Σ be a finite signature, let $\mathcal{G} \subseteq \mathsf{grAtom}(\Sigma)$ be a set of ground atoms, and let $\omega \in \Omega(\Sigma)$ be a possible world. Then, the *marginalization* of ω onto $\mathcal{G}$ is defined by

$$\omega_{|\mathcal{G}} = \bigwedge_{g \in \mathcal{G}\colon \omega \models g} g \wedge \bigwedge_{g \in \mathcal{G}\colon \omega \models \bar{g}} \bar{g}.$$

Every marginalized possible world $\omega_{|\mathcal{G}}$ is a partial possible world in $\Omega(\mathcal{G})$ and every partial possible world $\omega_{\mathcal{G}} \in \Omega(\mathcal{G})$ is the marginalization of some possible worlds in $\Omega(\Sigma)$. While the marginalization of possible worlds leads to unique partial possible worlds, the *expansion* of partial possible worlds in $\Omega(\mathcal{G})$ to possible worlds in $\Omega(\Sigma)$ is ambiguous if $\mathcal{G} \subsetneq \mathsf{grAtom}(\Sigma)$. With

$$\Omega_{\omega_{\mathcal{G}}}(\Sigma) = \{\omega \in \Omega(\Sigma) \mid \omega \models_{\Sigma} \omega_{\mathcal{G}}\}$$

we denote the set of all proper expansions of the partial possible world $\omega_{\mathcal{G}} \in \Omega(\mathcal{G})$. Of course, $\omega \in \Omega_{\omega_{|\mathcal{G}}}(\Sigma)$.

Possible worlds can be partitioned into partial possible worlds which can then be reassembled to the original possible world again if a partition of the ground atoms in $\mathsf{grAtom}(\Sigma)$ is given (cf. Definition A.1.1 in the appendix for a definition of partitions). This is useful to break down calculations based on possible worlds into local calculations. If $\{\mathcal{G}_1, \ldots, \mathcal{G}_m\}$ is a partition of $\mathsf{grAtom}(\Sigma)$, then, for $\omega \in \Omega(\Sigma)$, we have

$$\omega = \bigwedge_{i=1}^{m} \omega_{|\mathcal{G}_i}. \tag{3.2}$$

Equation (3.2) illustrates the usefulness of the representation of (partial) possible worlds as conjunctions which can simply be concatenated when combining them. We give an example.

Example 3.1.8

We recall the sentence

$$\phi'_{\mathsf{smk}} = \forall X.\exists_{\{Y \neq X\}} Y.\mathsf{Friends}(X, Y) \wedge \mathsf{Smokes}(Y)$$

from Example 2.2.9 which is defined over the signature Σ_{smk} (cf. Example 2.2.2(c)) with

$$\mathsf{Const}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{peter},\ \mathsf{paul},\ \mathsf{mary}\} \quad \text{and} \quad \mathsf{Pred}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{Friends}/2,\ \mathsf{Smokes}/1\}.$$

We abbreviate $F = \mathsf{Friends}$ and $S = \mathsf{Smokes}$. Since the interpretation of $F(X, X)$ for $X \in \{\mathsf{peter}, \mathsf{paul}, \mathsf{mary}\}$ is irrelevant for the evaluation of the sentence ϕ'_{smk}, we partition the set of ground atoms $\mathsf{grAtom}(\Sigma_{\mathsf{smk}})$ into $\mathcal{G}_1$ and $\mathcal{G}_2$ with

$$\begin{aligned}\mathcal{G}_1 = \{&F(\mathsf{peter}, \mathsf{paul}), F(\mathsf{peter}, \mathsf{mary}), F(\mathsf{paul}, \mathsf{peter}), F(\mathsf{paul}, \mathsf{mary}),\\ &F(\mathsf{mary}, \mathsf{peter}), F(\mathsf{mary}, \mathsf{paul}), S(\mathsf{peter}), S(\mathsf{paul}), S(\mathsf{mary})\},\\ \mathcal{G}_2 = \{&F(\mathsf{peter}, \mathsf{peter}), F(\mathsf{paul}, \mathsf{paul}), F(\mathsf{mary}, \mathsf{mary})\}.\end{aligned}$$

Only the partial possible worlds in $\Omega(\mathcal{G}_1)$ are relevant for the interpretation of ϕ'_{smk}. The truth assignment on the ground atoms in $\Omega(\mathcal{G}_2)$ does not affect the interpretation of ϕ'_{smk} instead. For example, the partial possible world $\omega_{\mathcal{G}_1} \in \Omega(\mathcal{G}_1)$ with

$$\begin{aligned}\omega_{\mathcal{G}_1} = &F(\mathsf{peter}, \mathsf{paul}) \land F(\mathsf{peter}, \mathsf{mary}) \land F(\mathsf{paul}, \mathsf{peter}) \land F(\mathsf{paul}, \mathsf{mary})\\ &\land F(\mathsf{mary}, \mathsf{peter}) \land F(\mathsf{mary}, \mathsf{paul}) \land S(\mathsf{peter}) \land \neg S(\mathsf{paul}) \land \neg S(\mathsf{mary})\end{aligned}$$

states that every two persons out of Peter, Paul, and Mary are friends and only Peter is a smoker, without making any statement about whether a person is a friend of herself. The partial possible world $\omega_{\mathcal{G}_1}$ induces the set of possible worlds $\Omega_{\omega_{\mathcal{G}_1}}(\Sigma_{\mathsf{smk}}) \subsetneq \Omega(\Sigma_{\mathsf{smk}})$ which coincide with $\omega_{\mathcal{G}_1}$ on $\mathcal{G}_1$ and specify whether $F(X, X)$ for $X \in \{\mathsf{peter}, \mathsf{paul}, \mathsf{mary}\}$ holds or not, in addition. In particular, we have $|\Omega_{\omega_{\mathcal{G}_1}}(\Sigma_{\mathsf{smk}})| = 2^3 = 8$. The restriction to relevant partial possible worlds has a positive impact on computational costs in a natural way as it lowers the exponent in (3.1).

Next, we evaluate conditionals with respect to possible worlds based on so-called *counting functions* which generalize the trivalent evaluation of closed conditionals.

▶ Evaluation of Closed Conditionals and Counting Functions

We discuss how to evaluate relational conditionals which possibly mention free variables based on possible worlds. The basic idea is to count how many instances of a conditional are verified, falsified or not applicable. This count-based evaluation extends the three-valued evaluation of closed conditionals from [de Finetti, 2017] and will be of importance when developing probabilistic semantics of conditionals that are equipped with probability values in Section 3.3.

ω	$\delta_{\mathsf{bow}}(\omega)$	ω	$\delta_{\mathsf{bow}}(\omega)$
BRS	1	$\overline{B}RS$	0
$BR\overline{S}$	u	$\overline{B}R\overline{S}$	u
$B\overline{R}S$	u	$\overline{B}\overline{R}S$	u
$B\overline{R}\overline{S}$	u	$\overline{B}\overline{R}\overline{S}$	u

Table 3.1: Evaluation of the closed conditional $\delta_{\mathsf{bow}} = (\mathsf{Rainbow}|\mathsf{Sunshine} \wedge \mathsf{Rain})$ with respect to the possible worlds in $\Omega(\Sigma_{\mathsf{bow}})$ (cf. Example 3.1.10). We use the abbreviations $B = \mathsf{Rainbow}$, $S = \mathsf{Sunshine}$, and $R = \mathsf{Rain}$.

Definition 3.1.9: Evaluation of Closed Conditionals

(cf. [de Finetti, 2017])

Let Σ be a finite signature, let $(\psi|\phi) \in \mathcal{RCL}(\Sigma)$ be a closed conditional, and let $\omega \in \Omega(\Sigma)$ be a possible world. Then, we define the *evaluation* of $(\psi|\phi)$ in ω by

$$(\psi|\phi)(\omega) = \begin{cases} 1 & \text{if} \quad \omega \models_\Sigma \phi\psi & \text{(verification)} \\ 0 & \text{if} \quad \omega \models_\Sigma \phi\overline{\psi} & \text{(falsification)} \\ u & \text{if} \quad \omega \models_\Sigma \overline{\phi} & \text{(non-applicability)} \end{cases}.$$

A closed conditional $(\psi|\phi) \in \mathcal{RCL}(\Sigma)$ is *verified* in a possible world $\omega \in \Omega(\Sigma)$ iff both the premise ϕ and the conclusion ψ of the conditional are true in ω. It is *falsified* if the premise is true and the conclusion is false, and it is *not applicable* if the premise is false.

Example 3.1.10

We consider the closed conditional $\delta_{\mathsf{bow}} = (\mathsf{Rainbow}|\mathsf{Sunshine} \wedge \mathsf{Rain})$

"The concurrence of sunshine and rain usually causes rainbows."

over the signature $\Sigma_{\mathsf{bow}} = (\emptyset, \mathsf{Pred}_{\Sigma_{\mathsf{bow}}})$ (cf. Example 2.2.2(a)) with

$$\mathsf{Pred}_{\Sigma_{\mathsf{bow}}} = \{\mathsf{Sunshine}/0,\ \mathsf{Rain}/0,\ \mathsf{Rainbow}/0\}.$$

The evaluation of the conditional δ_{bow} with respect to the possible worlds in $\Omega(\Sigma_{\mathsf{bow}})$ is shown in Table 3.1.

The three-valued evaluation of closed conditionals can be extended to open conditionals by counting the number of instances of the conditionals that are verified, falsified, or (not) applicable within a possible world $\omega \in \Omega(\Sigma)$.

Definition 3.1.11: Counting Functions

(based on [Kern-Isberner and Thimm, 2012])

Let Σ be a finite signature, let $\delta = (\psi|\phi)_{\mathsf{CS}}$ be a conditional from $\mathcal{RCL}(\Sigma)$, and let $\omega \in \Omega(\Sigma)$ be a possible world. Then, we define the *counting functions*

$$\begin{aligned}
\mathsf{ver}_{\Sigma,\delta}(\omega) &= |\{(\psi'|\phi') \in \mathsf{Inst}_\Sigma(\delta) \mid \omega \models_\Sigma \phi'\psi'\}|, && (\#\,\text{verifications})\\
\mathsf{fal}_{\Sigma,\delta}(\omega) &= |\{(\psi'|\phi') \in \mathsf{Inst}_\Sigma(\delta) \mid \omega \models_\Sigma \phi'\overline{\psi'}\}|, && (\#\,\text{falsifications})\\
\mathsf{app}_{\Sigma,\delta}(\omega) &= |\{(\psi'|\phi') \in \mathsf{Inst}_\Sigma(\delta) \mid \omega \models_\Sigma \phi'\}|, && (\#\,\text{applicabilities})\\
\mathsf{napp}_{\Sigma,\delta}(\omega) &= |\{(\psi'|\phi') \in \mathsf{Inst}_\Sigma(\delta) \mid \omega \not\models_\Sigma \phi'\}|. && (\#\,\text{non-applicabilities})
\end{aligned}$$

The counting functions in Definition 3.1.11 map possible worlds to natural numbers (including zero). Every two of the counts $\mathsf{ver}_{\Sigma,\delta}(\omega)$, $\mathsf{fal}_{\Sigma,\delta}(\omega)$, and $\mathsf{app}_{\Sigma,\delta}(\omega)$ determine the third one. More precisely, we have the identity

$$\mathsf{app}_{\Sigma,\delta}(\omega) = \mathsf{ver}_{\Sigma,\delta}(\omega) + \mathsf{fal}_{\Sigma,\delta}(\omega)$$

for $\omega \in \Omega(\Sigma)$. The count $\mathsf{napp}_{\Sigma,\delta}(\omega)$ can be derived from $\mathsf{app}_{\Sigma,\delta}(\omega)$ by

$$\mathsf{napp}_{\Sigma,\delta}(\omega) = |\mathsf{Inst}_\Sigma(\delta)| - \mathsf{app}_{\Sigma,\delta}(\omega).$$

Consequently, only two of the above-mentioned counts $\mathsf{ver}_{\Sigma,\delta}(\omega)$, $\mathsf{fal}_{\Sigma,\delta}(\omega)$, and $\mathsf{app}_{\Sigma,\delta}(\omega)$ are necessary to represent the overall evaluation of an open conditional with respect to the possible world ω. Hereby, note that $\mathsf{Inst}_\Sigma(\delta)$ is independent of ω. From another perspective, we have

$$|\mathsf{Inst}_\Sigma(\delta)| = \mathsf{ver}_{\Sigma,\delta}(\omega) + \mathsf{fal}_{\Sigma,\delta}(\omega) + \mathsf{napp}_{\Sigma,\delta}(\omega)$$

which reflects that each instance of δ is either verified, falsified, or not applicable in ω. Note that different instances of the same open conditional can be evaluated differently by a possible world. For example, one instance can be verified while another instance is falsified.

Example 3.1.12

We continue the previous Examples 3.1.8 and 3.1.3 and consider the signature $\Sigma_{\mathsf{smk}} = (\mathsf{Const}_{\Sigma_{\mathsf{smk}}}, \mathsf{Pred}_{\Sigma_{\mathsf{smk}}})$ (cf. Example 2.2.2(c)) with

$$\mathsf{Const}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{peter},\ \mathsf{paul},\ \mathsf{mary}\} \quad \text{and} \quad \mathsf{Pred}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{Friends}/2,\ \mathsf{Smokes}/1\}.$$

We exemplarily evaluate the open conditional

$$\delta_{\mathsf{smk}} = (\mathsf{Smokes}(X)|\mathsf{Friends}(X,Y) \wedge \mathsf{Smokes}(Y))_{\{X \neq Y\}}$$

with respect to the possible worlds in $\Omega_\omega(\Sigma)$ where ω is the partial possible world (we abbreviate $F = \mathsf{Friends}$ and $S = \mathsf{Smokes}$)

$$\begin{aligned}\omega = {} & F(\mathsf{peter},\mathsf{paul}) \wedge F(\mathsf{peter},\mathsf{mary}) \wedge F(\mathsf{paul},\mathsf{peter}) \wedge F(\mathsf{paul},\mathsf{mary}) \\ & \wedge F(\mathsf{mary},\mathsf{peter}) \wedge F(\mathsf{mary},\mathsf{paul}) \wedge S(\mathsf{peter}) \wedge \neg S(\mathsf{paul}) \wedge \neg S(\mathsf{mary}).\end{aligned}$$

Note that we have already noticed that the evaluation of δ_{smk} is the same in all possible worlds from $\Omega_\omega(\Sigma_{\mathsf{smk}})$ because $\Omega_\omega(\Sigma_{\mathsf{smk}}) = \Omega(\mathcal{G}_1)$ where $\mathcal{G}_1$ is the respective set of atoms from Example 3.1.8 and the evaluation of the ground atoms in $\mathcal{G}_2 = \mathsf{grAtom}(\Sigma) \setminus \mathcal{G}_1$ is irrelevant for the evaluation of δ_{smk} since no instance of δ_{smk} mentions any ground atom $F(X,X)$ with $X \in \{\mathsf{peter},\mathsf{paul},\mathsf{mary}\}$.

Because Peter is the only smoker in the partial possible world ω, no instance of δ_{smk} is verified in any possible world $\omega' \in \Omega_\omega(\Sigma_{\mathsf{smk}})$. Actually, this is because the verification of any instance of δ_{smk} requires at least two smokers because the variables X and Y in δ_{smk} have to be instantiated with different constants. Further, the instances

$$\begin{gathered}(\mathsf{Smokes}(\mathsf{paul})|\mathsf{Friends}(\mathsf{paul},\mathsf{peter}) \wedge \mathsf{Smokes}(\mathsf{peter})), \\ (\mathsf{Smokes}(\mathsf{mary})|\mathsf{Friends}(\mathsf{mary},\mathsf{peter}) \wedge \mathsf{Smokes}(\mathsf{peter})),\end{gathered}$$

of δ_{smk} are falsified in all possible worlds $\omega' \in \Omega_\omega(\Sigma_{\mathsf{smk}})$ and all other instances are not applicable. Consequently, we observe the following counts:

$$\begin{aligned}\mathsf{ver}_{\Sigma_{\mathsf{smk}},\delta_{\mathsf{smk}}}(\omega') &= 0, & \mathsf{fal}_{\Sigma_{\mathsf{smk}},\delta_{\mathsf{smk}}}(\omega') &= 2, \\ \mathsf{app}_{\Sigma_{\mathsf{smk}},\delta_{\mathsf{smk}}}(\omega') &= 2, & \mathsf{napp}_{\Sigma_{\mathsf{smk}},\delta_{\mathsf{smk}}}(\omega') &= 4.\end{aligned}$$

In the possible worlds in which Peter, Paul, and Mary are friends and smokers, i.e., in all possible worlds $\hat{\omega}' \in \Omega_{\hat{\omega}}(\Sigma_{\mathsf{smk}})$ which are expansions of

$$\begin{aligned}\hat{\omega} = {} & F(\mathsf{peter},\mathsf{paul}) \wedge F(\mathsf{peter},\mathsf{mary}) \wedge F(\mathsf{paul},\mathsf{peter}) \wedge F(\mathsf{paul},\mathsf{mary}) \\ & \wedge F(\mathsf{mary},\mathsf{peter}) \wedge F(\mathsf{mary},\mathsf{paul}) \wedge S(\mathsf{peter}) \wedge S(\mathsf{paul}) \wedge S(\mathsf{mary}),\end{aligned}$$

we have

$$\begin{aligned}\mathsf{ver}_{\Sigma_{\mathsf{smk}},\delta_{\mathsf{smk}}}(\hat{\omega}') &= 6, & \mathsf{fal}_{\Sigma_{\mathsf{smk}},\delta_{\mathsf{smk}}}(\hat{\omega}') &= 0, \\ \mathsf{app}_{\Sigma_{\mathsf{smk}},\delta_{\mathsf{smk}}}(\hat{\omega}') &= 6, & \mathsf{napp}_{\Sigma_{\mathsf{smk}},\delta_{\mathsf{smk}}}(\hat{\omega}') &= 0.\end{aligned}$$

As closed conditionals $\delta \in \mathcal{RCL}(\Sigma)$ have only one instance, namely δ itself, i.e., $\mathsf{Inst}_{\Sigma}(\delta) = \{\delta\}$, we have the following straightforward characterization of the evaluation of closed conditionals.

Proposition 3.1.13: Evaluation of Closed Conditionals
Let Σ be a finite signature, let $\delta \in \mathcal{RCL}(\Sigma)$ be a closed conditional, and let $\omega \in \Omega(\Sigma)$ be a possible world. Then,

$$\delta(\omega) = \begin{cases} 1 & \text{if} \quad \mathsf{ver}_{\Sigma,\delta}(\omega) = 1 \\ 0 & \text{if} \quad \mathsf{fal}_{\Sigma,\delta}(\omega) = 1 \\ u & \text{if} \quad \mathsf{napp}_{\Sigma,\delta}(\omega) = 1 \end{cases} . \tag{3.3}$$

Proof. The proposition follows directly from the three-valued evaluation of closed conditionals and the definition of the counting functions. □

Intuitively, if one believes in a conditional $\delta \in \mathcal{RCL}(\Sigma)$, i.e., if one accepts the assumption that the premise of δ usually implies its consequence, then one should also believe that possible worlds in which the number of verifications of δ outweigh the number of falsifications of δ tend to be more likely representations of the real world than possible worlds in which this is not true. In this sense, in Example 3.1.12, the possible worlds in $\Omega_{\hat{\omega}}(\Sigma_{\mathsf{smk}})$ are more likely formalizations of the real world than the possible worlds in $\Omega_{\omega}(\Sigma_{\mathsf{smk}})$, provided that the conditional δ_{smk} is believed. This motivates the counting of verifications and falsifications (and the further counts) of a conditional within single possible worlds.

Beyond that, Example 3.1.10 allows for drawing attention to the fact that counting the verifications and falsifications of a conditional $\delta \in \mathcal{RCL}(\Sigma)$ can also be understood in another way than captured in $\mathsf{ver}_{\Sigma,\delta}(\omega)$ and $\mathsf{fal}_{\Sigma,\delta}(\omega)$. Instead of counting the verified and falsified instances of δ within a possible world, one can also count the number of possible worlds in which an instance, or generally a closed conditional, is verified or falsified (resp. not applicable). For instance, in Example 3.1.10, the closed conditional δ_{bow} is verified and falsified in only one possible world and not applicable in six possible worlds. This leads to another way of counting which is orthogonal to $\mathsf{ver}_{\Sigma,\delta}(\omega)$ and $\mathsf{fal}_{\Sigma,\delta}(\omega)$.

We will revisit both views on counting verifications and falsifications when we have extended qualitative conditionals by probabilities and discuss the *aggregating semantics* of probabilistic conditionals in Section 3.4. Beforehand, we recall *conditional structures* from [Kern-Isberner, 2001a] as a compact and useful representation of these counts.

3.2 Conditional Structures

In this section, we discuss *conditional structures* [Kern-Isberner, 2001a] as a formal representation of the evaluation of conditionals in possible worlds. Thereafter, we extend the notion of conditional structures to sets of possible worlds and discuss first strategies for an efficient calculation of conditional structures. We close this section by mentioning the concept of *conditional equivalence*.

▶ Conditional Structures of Possible Worlds

Conditional structures are algebraic objects which reflect the number of verified and falsified instances of the conditionals in a finite set $\Delta \subseteq \mathcal{RCL}(\Sigma)$ within a possible world. Mathematically, they are elements of the finitely generated *free Abelian group*

$$\mathcal{G}(\Delta) = (\mathcal{G}_G(\Delta), \cdot, 1)$$

with *generating set*

$$\mathcal{G}_G(\Delta) = \{\mathsf{a}_i^+ \mid i = 1, \ldots, n\} \cup \{\mathsf{a}_i^- \mid i = 1, \ldots, n\},$$

where $n = |\Delta|$. We refer to Definition A.1.3 in the appendix for a formal definition of a mathematical *group* (cf. also [Lang, 2002]). A group is called *Abelian* if the operation of the group, here the multiplication $\cdot$, is *commutative*. The fact that the Abelian group $\mathcal{G}(\Delta)$ is *free* means that every element in $\mathcal{G}(\Delta)$ can be written as a unique $\mathbb{Z}$-linear combination of its *generators* in $\mathcal{G}_G(\Delta)$ modulo commutation. Hence, because $\mathcal{G}(\Delta)$ is multiplicative, elements in $\mathcal{G}(\Delta)$ are of the form $\prod_{i=1}^{n} (\mathsf{a}_i^+)^{m_i} \cdot (\mathsf{a}_i^-)^{k_i}$ with $m_i, k_i \in \mathbb{Z}$ for $i = 1, \ldots, n$, whereby $1 = \prod_{i=1}^{n} (\mathsf{a}_i^+)^0 \cdot (\mathsf{a}_i^-)^0$ is the identity element in $\mathcal{G}(\Delta)$.

Definition 3.2.1: Conditional Structure *[Kern-Isberner, 2001a]*

Let Σ be a finite signature, let $\Delta \subseteq \mathcal{RCL}(\Sigma)$ be a finite set of conditionals, say $\Delta = \{\delta_1, \ldots, \delta_n\}$, and let $\omega \in \Omega(\Sigma)$ be a possible world. Then, the *conditional structure* of ω with respect to Δ is

$$\sigma_{\Sigma,\Delta}(\omega) = \prod_{i=1}^{n} (\mathsf{a}_i^+)^{\mathsf{ver}_{\Sigma,\delta_i}(\omega)} \cdot (\mathsf{a}_i^-)^{\mathsf{fal}_{\Sigma,\delta_i}(\omega)}.$$

The generators a_i^+ and a_i^- of $\mathcal{G}(\Delta)$ tell us whether the i-th conditional from Δ is verified (a_i^+) or falsified (a_i^-) in the possible world ω, namely by occurring or not occurring in $\sigma_{\Sigma,\Delta}(\omega)$. From the exponents of these generators we can read *how often* the respective conditional is verified resp. falsified. We sometimes use the notation

$$\sigma_{\Sigma,\delta}(\omega) = (\mathsf{a}_\delta^+)^{\mathsf{ver}_{\Sigma,\delta}(\omega)} \cdot (\mathsf{a}_\delta^-)^{\mathsf{fal}_{\Sigma,\delta}(\omega)}$$

for conditional structures with respect to single conditionals $\delta \in \mathcal{RCL}(\Sigma)$, too. The advantage of considering conditional structures as elements of a free Abelian group is the possibility to apply mathematical operations to conditional structures which enables us to calculate on them. For example, it is possible to multiply conditional structures of several conditionals so that, in particular,

$$\sigma_{\Sigma,\Delta}(\omega) = \prod_{\delta \in \Delta} \sigma_{\Sigma,\delta}(\omega)$$

holds. Efficient calculations of conditional structures will be of great importance when we address lifted probabilistic inference at maximum entropy in the following chapters.

Recall that the multiplication in the Abelian group $\mathcal{G}(\Delta)$ is commutative. Because of this, the order of the conditionals which is induced by the enumeration in Definition 3.2.1 is irrelevant and just used for simpler referencing. The identity element 1 reflects that no conditional in Δ is applicable. Further, every element in $\mathcal{G}(\Delta)$ has its inverse element in $\mathcal{G}(\Delta)$ albeit the inverse elements neither occur in the conditional structures directly nor play a role in this thesis. They can be useful when drawing conditional inferences on an abstract, algebraic level, for example. Then, quotients of conditional structures can be calculated and possibly canceled against each other (cf. [Kern-Isberner, 2001a]).

When we enumerate the elements in Δ, i.e., when we refer to the conditionals in Δ with δ_i for $i = 1, \ldots, n$, then we also use the notation $\sigma_{\Sigma,i}(\omega)$ for $\sigma_{\Sigma,\delta_i}(\omega)$ in order to avoid double indices. Analogously, we write $\mathsf{ver}_{\Sigma,i}(\omega)$ and $\mathsf{fal}_{\Sigma,i}(\omega)$ instead of $\mathsf{ver}_{\Sigma,\delta_i}(\omega)$ and $\mathsf{fal}_{\Sigma,\delta_i}(\omega)$. That is, we refer to the i-th conditional δ_i in Δ by its index i for the sake of simplicity. We will follow this schema in the rest of this thesis without explicitly mentioning it again. For example, we write $\mathsf{app}_{\Sigma,i}(\omega)$ instead of $\mathsf{app}_{\Sigma,\delta_i}(\omega)$, too. The other way around, we sometimes use the notations $\mathsf{a}_{\delta_i}^+$ and $\mathsf{a}_{\delta_i}^-$ instead of a_i^+ and a_i^-.

Example 3.2.2

We consider the signature $\Sigma_{\mathsf{smk}} = (\mathsf{Const}_{\Sigma_{\mathsf{smk}}}, \mathsf{Pred}_{\Sigma_{\mathsf{smk}}})$ with

$$\mathsf{Const}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{peter},\ \mathsf{paul},\ \mathsf{mary}\} \quad \text{and} \quad \mathsf{Pred}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{Friends}/2,\ \mathsf{Smokes}/1\}$$

and the conditional

$$\delta_{\mathsf{smk}} = (\mathsf{Smokes}(X)|\mathsf{Friends}(X,Y) \wedge \mathsf{Smokes}(Y))_{\{X \neq Y\}}$$

as in Example 3.1.3. According to Example 3.1.12, we have

$$\begin{aligned}\sigma_{\Sigma_{\mathsf{smk}},\delta_{\mathsf{smk}}}(\omega') &= (\mathsf{a}^{+}_{\delta_{\mathsf{smk}}})^{0} \cdot (\mathsf{a}^{-}_{\delta_{\mathsf{smk}}})^{2} = (\mathsf{a}^{-}_{\delta_{\mathsf{smk}}})^{2},\\ \sigma_{\Sigma_{\mathsf{smk}},\delta_{\mathsf{smk}}}(\hat{\omega}') &= (\mathsf{a}^{+}_{\delta_{\mathsf{smk}}})^{6} \cdot (\mathsf{a}^{-}_{\delta_{\mathsf{smk}}})^{0} = (\mathsf{a}^{+}_{\delta_{\mathsf{smk}}})^{6},\end{aligned}$$

where (a) ω' is any possible world in which Peter, Paul, and Mary are friends and only Peter is a smoker, and where (b) $\hat{\omega}'$ is any possible world in which all three persons are friends as well as smokers.

To illustrate conditional structures with respect to sets of conditionals, we consider the additional conditional

$$\delta_{\mathsf{frnd}} = (\mathsf{Friends}(Y,X)|\mathsf{Friends}(X,Y))_{X \neq Y}$$

which states that being friends is usually a symmetric relation and define $\Delta_{\mathsf{smk}} = \{\delta_{\mathsf{smk}}, \delta_{\mathsf{frnd}}\}$. The conditional δ_{frnd} has six proper instances:

$$\begin{aligned}\mathsf{Inst}_{\Sigma_{\mathsf{smk}}}(\delta_{\mathsf{frnd}}) = \{&(\mathsf{Friends}(\mathsf{peter},\mathsf{paul})|\mathsf{Friends}(\mathsf{paul},\mathsf{peter})),\\ &(\mathsf{Friends}(\mathsf{peter},\mathsf{mary})|\mathsf{Friends}(\mathsf{mary},\mathsf{peter})),\\ &(\mathsf{Friends}(\mathsf{paul},\mathsf{peter})|\mathsf{Friends}(\mathsf{peter},\mathsf{paul})),\\ &(\mathsf{Friends}(\mathsf{paul},\mathsf{mary})|\mathsf{Friends}(\mathsf{mary},\mathsf{paul})),\\ &(\mathsf{Friends}(\mathsf{mary},\mathsf{peter})|\mathsf{Friends}(\mathsf{peter},\mathsf{mary})),\\ &(\mathsf{Friends}(\mathsf{mary},\mathsf{paul})|\mathsf{Friends}(\mathsf{paul},\mathsf{mary}))\}.\end{aligned}$$

All of them are verified in the above-mentioned possible worlds ω' and $\hat{\omega}'$ and, thus, we have

$$\begin{aligned}\sigma_{\Sigma_{\mathsf{smk}},\Delta_{\mathsf{smk}}}(\omega') &= \sigma_{\Sigma_{\mathsf{smk}},\delta_{\mathsf{smk}}}(\omega') \cdot \sigma_{\Sigma_{\mathsf{smk}},\delta_{\mathsf{frnd}}}(\omega') = (\mathsf{a}^{-}_{\delta_{\mathsf{smk}}})^{2} \cdot (\mathsf{a}^{+}_{\delta_{\mathsf{frnd}}})^{6},\\ \sigma_{\Sigma_{\mathsf{smk}},\Delta_{\mathsf{smk}}}(\hat{\omega}') &= \sigma_{\Sigma_{\mathsf{smk}},\delta_{\mathsf{smk}}}(\hat{\omega}') \cdot \sigma_{\Sigma_{\mathsf{smk}},\delta_{\mathsf{frnd}}}(\hat{\omega}') = (\mathsf{a}^{+}_{\delta_{\mathsf{smk}}})^{6} \cdot (\mathsf{a}^{+}_{\delta_{\mathsf{frnd}}})^{6}.\end{aligned}$$

Now, we extend conditional structures to sets of possible worlds.

▶ Conditional Structures of Sets of Possible Worlds

Conditional structures of possible worlds allow us to *multiply* formal representatives of the evaluation of conditionals with respect to *single* possible worlds, namely a_i^+ and a_i^- for $i = 1, \ldots, n$. Later on, we will see that it is also useful to *add* conditional structures of *several* possible worlds, i.e., to build sums like $\sigma_{\Sigma,\Delta}(\omega) + \sigma_{\Sigma,\Delta}(\omega')$ for $\omega, \omega' \in \Omega(\Sigma)$. In order to be able to do so, we extend the free Abelian group $\mathcal{G}(\Delta)$ to the *zerosumfree Abelian semiring*

$$\mathcal{S}(\Delta) = (\mathcal{G}_G(\Delta), +, \cdot, 1, 0), \tag{3.4}$$

where $\mathcal{G}_G(\Delta)$ is the generating set of $\mathcal{G}(\Delta)$ which now serves as a generating set of $\mathcal{S}(\Delta)$. Elements in $\mathcal{S}(\Delta)$ are of the form $\sum_{h=1}^{l} \prod_{i=1}^{n} (\mathsf{a}_i^+)^{m_i(h)} \cdot (\mathsf{a}_i^-)^{k_i(h)}$ with $l \in \mathbb{N}_0$ and $m_i(h), k_i(h) \in \mathbb{Z}$ for $i = 1, \ldots, n$ and $h = 1, \ldots, l$. In case of $l = 0$, we observe an empty sum which refers to the identity element regarding the addition $+$ in $\mathcal{S}(\Delta)$, and which is denoted by 0. Basically, $\mathcal{S}(\Delta)$ is $\mathcal{G}(\Delta)$ with the ability to sum up elements from $\mathcal{G}(\Delta)$. Calculations in $\mathcal{S}(\Delta)$ proceed intuitively except for the fact that there is no subtraction in $\mathcal{S}(\Delta)$, i.e., there are no inverse elements with respect to the additive operation $+$. The *zerosumfreeness* of $\mathcal{S}(\Delta)$ means that for two elements $s, s' \in \mathcal{S}(\Delta)$ it directly follows from $s + s' = 0$ that both s and s' must be 0. Please see Definition A.1.4 in the appendix for a definition of Abelian semirings and also [Golan, 1999] for a more in-depth introduction to semirings and the concept of zerosumfreeness.

The semiring $\mathcal{S}(\Delta)$ allows us to generalize conditional structures in the following sense.

Definition 3.2.3: Conditional Structure of Sets of Possible Worlds
Let Σ be a finite signature, let $\Delta \subseteq \mathcal{RCL}(\Sigma)$ be a finite set of conditionals, and let $\Omega' \subseteq \Omega(\Sigma)$ be a set of possible worlds. Then, the *conditional structure* of Ω' with respect to Δ is

$$\sigma_{\Sigma,\Delta}(\Omega') = \sum_{\omega \in \Omega'} \sigma_{\Sigma,\Delta}(\omega) = \sum_{\omega \in \Omega'} \prod_{\delta \in \Delta} \sigma_{\Sigma,\delta}(\omega). \tag{3.5}$$

The reason why we introduce conditional structures of sets of possible worlds is that we aim on calculating conditional structures for several possible worlds simultaneously. Based on (3.5), it is sometimes possible to find a very condensed representation of the conditional structures of sets of possible worlds by making use of the semiring laws, in particular, distributivity. The crucial task here is to factorize $\sigma_{\Sigma,\Delta}(\Omega')$ which means a commutation of the sum and the product in (3.5). The next examples illustrate this idea very impressively.

ω	$\sigma_{\Sigma_i,\delta_1}(\omega)$	$\sigma_{\Sigma_i,\delta_2}(\omega)$	$\sigma_{\Sigma_i,\delta_3}(\omega)$
$B(c_i)F(c_i)P(c_i)$	a_1^+	a_2^+	a_3^-
$B(c_i)F(c_i)\overline{P(c_i)}$	a_1^+	a_2^+	1
$B(c_i)\overline{F(c_i)}P(c_i)$	a_1^-	a_2^+	a_3^+
$B(c_i)\overline{F(c_i)}\,\overline{P(c_i)}$	a_1^-	a_2^+	1
$\overline{B(c_i)}F(c_i)P(c_i)$	1	a_2^-	a_3^-
$\overline{B(c_i)}F(c_i)\overline{P(c_i)}$	1	a_2^+	1
$\overline{B(c_i)}\,\overline{F(c_i)}P(c_i)$	1	a_2^-	a_3^+
$\overline{B(c_i)}\,\overline{F(c_i)}\,\overline{P(c_i)}$	1	a_2^+	1

Table 3.2: Conditional structures of the possible worlds in $\Omega(\Sigma_i)$ with respect to the conditionals $\delta_1 = (F(c_i)|B(c_i))$, $\delta_2 = (P(c_i) \Rightarrow B(c_i)|\top)$, and $\delta_3 = (\neg F(c_i)|P(c_i))$ from Δ_i (cf. Example 3.2.4), where the abbreviations $B = \mathsf{Bird}$, $F = \mathsf{Flies}$, and $P = \mathsf{Penguin}$ apply.

Example 3.2.4
We consider the set of conditionals

$$\begin{aligned}\Delta_{\mathsf{bfp}} = \{&(\mathsf{Flies}(X)|\mathsf{Bird}(X)),\\ &(\mathsf{Penguin}(X) \Rightarrow \mathsf{Bird}(X)|\top),\\ &(\neg\mathsf{Flies}(X)|\mathsf{Penguin}(X))\}\end{aligned}$$

defined over the signature $\Sigma_{\mathsf{bfp}} = (\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}, \mathsf{Pred}_{\Sigma_{\mathsf{bfp}}})$ from Example 2.2.2(b). It is

$$\mathsf{Pred}_{\Sigma_{\mathsf{bfp}}} = \{\mathsf{Bird}/1, \mathsf{Flies}/1, \mathsf{Penguin}/1\},$$

and we assume $\mathsf{Const}_{\Sigma_{\mathsf{bfp}}} = \{c_1, c_2, c_3\}$ so that $|\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}| = 3$. Because $|\Omega(\Sigma_{\mathsf{bfp}})| = (2^3)^3 = 512$, computing the conditional structures of all possible worlds in $\Omega(\Sigma_{\mathsf{bfp}})$ independently is quite time-consuming—at least when computing them by hand—even for this small number of constants. However, for some reasoning tasks it turns out that it is sufficient to compute the conditional structures simultaneously in one single expression $\sigma_{\Sigma_{\mathsf{bfp}},\Delta_{\mathsf{bfp}}}(\Omega(\Sigma_{\mathsf{bfp}}))$ which can be obtained much easier. Then, one loses the information which conditional structure refers to which possible world, but this information is not always necessary. In this example, we have

$$\begin{aligned}\sigma_{\Sigma_{\mathsf{bfp}},\Delta_{\mathsf{bfp}}}(\Omega(\Sigma_{\mathsf{bfp}})) &= \sum_{\omega\in\Omega(\Sigma_{\mathsf{bfp}})}\prod_{\delta\in\Delta_{\mathsf{bfp}}}(\mathsf{a}_{\delta}^{+})^{\mathsf{ver}_{\Sigma_{\mathsf{bfp}},\delta}(\omega)}\cdot(\mathsf{a}_{\delta}^{-})^{\mathsf{fal}_{\Sigma_{\mathsf{bfp}},\delta}(\omega)}\\ &= \sum_{\omega\in\Omega(\Sigma_{\mathsf{bfp}})}\prod_{\delta\in\Delta_{\mathsf{bfp}}}\prod_{\delta'\in\mathsf{Inst}_{\Sigma_{\mathsf{bfp}}}(\delta)}(\mathsf{a}_{\delta'}^{+})^{\mathsf{ver}_{\Sigma_{\mathsf{bfp}},\delta'}(\omega)}\cdot(\mathsf{a}_{\delta'}^{-})^{\mathsf{fal}_{\Sigma_{\mathsf{bfp}},\delta'}(\omega)}.\end{aligned}$$

The ground instances $\delta'\in\mathsf{Inst}_{\Sigma_{\mathsf{bfp}}}(\delta)$ of the conditionals $\delta\in\Delta_{\mathsf{bfp}}$ are obtained by substituting the (only) free variable X by any constant from $\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}$ so that they make use of exactly one of the three pairwise disjoint sets of ground atoms

$$\mathcal{G}_i = \{\mathsf{Bird}(c_i),\mathsf{Flies}(c_i),\mathsf{Penguin}(c_i)\},\qquad i=1,2,3,$$

which partition $\mathsf{grAtom}(\Sigma_{\mathsf{bfp}})$. In particular, the evaluation of the ground instance of the conditionals $\delta\in\Delta_{\mathsf{bfp}}$ which make use of ground atoms from $\mathcal{G}_i$ and which we name δ_{c_i}, $i=1,2,3$ from now on, is independent of the truth assignment on $\mathcal{G}_j$ with $j\neq i$. Therewith, it follows that

$$\begin{aligned}&\sigma_{\Sigma_{\mathsf{bfp}},\Delta_{\mathsf{bfp}}}(\Omega(\Sigma_{\mathsf{bfp}}))\\ &= \sum_{\omega_1\in\Omega(\Sigma_1)}\sum_{\omega_2\in\Omega(\Sigma_2)}\sum_{\omega_3\in\Omega(\Sigma_3)}\prod_{\delta\in\Delta_{\mathsf{bfp}}}\prod_{i=1}^{3}(\mathsf{a}_{\delta_{c_i}}^{+})^{\mathsf{ver}_{\Sigma_i,\delta_{c_i}}(\omega_i)}\cdot(\mathsf{a}_{\delta_{c_i}}^{-})^{\mathsf{fal}_{\Sigma_i,\delta_{c_i}}(\omega_i)}\\ &= \sum_{\omega_1\in\Omega(\Sigma_1)}\sum_{\omega_2\in\Omega(\Sigma_2)}\sum_{\omega_3\in\Omega(\Sigma_3)}\prod_{i=1}^{3}\prod_{\delta\in\Delta_{\mathsf{bfp}}}(\mathsf{a}_{\delta_{c_i}}^{+})^{\mathsf{ver}_{\Sigma_i,\delta_{c_i}}(\omega_i)}\cdot(\mathsf{a}_{\delta_{c_i}}^{-})^{\mathsf{fal}_{\Sigma_i,\delta_{c_i}}(\omega_i)},\end{aligned}$$

where $\Sigma_i = (\{c_i\},\mathsf{Pred}_{\mathsf{bfp}})$ for $i=1,2,3$. With $\Delta_i=\{\delta_{c_i}\mid\delta\in\Delta_{\mathsf{bfp}}\}$ for $i=1,2,3$, we further obtain

$$\begin{aligned}&\sigma_{\Sigma_{\mathsf{bfp}},\Delta_{\mathsf{bfp}}}(\Omega(\Sigma_{\mathsf{bfp}}))\\ &= \sum_{\omega_1\in\Omega(\Sigma_1)}\sum_{\omega_2\in\Omega(\Sigma_2)}\sum_{\omega_3\in\Omega(\Sigma_3)}\prod_{i=1}^{3}\sigma_{\Sigma_i,\Delta_i}(\omega_i)\\ &= \sum_{\omega_1\in\Omega(\Sigma_1)}\sigma_{\Sigma_1,\Delta_1}(\omega_1)\sum_{\omega_2\in\Omega(\Sigma_2)}\sigma_{\Sigma_2,\Delta_2}(\omega_2)\sum_{\omega_3\in\Omega(\Sigma_3)}\sigma_{\Sigma_3,\Delta_3}(\omega_3)\\ &= \prod_{i=1}^{3}\sum_{\omega_i\in\Omega(\Sigma_i)}\sigma_{\Sigma_i,\Delta_i}(\omega_i).\end{aligned}$$

The last rearrangement of $\sigma_{\Sigma_{\mathsf{bfp}},\Delta_{\mathsf{bfp}}}(\Omega(\Sigma_{\mathsf{bfp}}))$ leads to the desired permutation of the sum and the product in $\sigma_{\Sigma_{\mathsf{bfp}},\Delta_{\mathsf{bfp}}}(\Omega(\Sigma_{\mathsf{bfp}}))$ and, therewith, to a "localization" of the calculation of possible worlds. In contrast to the initial situation in which 512 conditional structures of possible worlds had to be calculated, now $3 \cdot (2^3) = 24$ computations of conditional structures remain which, furthermore, refer to much smaller signatures (Σ_i) and to sets of closed conditionals (Δ_i) than the direct computations. Hence, computations simplify drastically.

Moreover, the signatures Σ_i and the sets of conditionals Δ_i are *isomorphic* which means that they are the same for $i = 1, 2, 3$ up to a renaming of the constants. This again means that we do not have to compute $\sigma_{\Sigma_i,\Delta_i}(\omega_i)$ with respect to all indices $i = 1, 2, 3$, but it suffices to compute this expression once for any index i because $\sigma_{\Sigma_i,\Delta_i}(\omega_i) = \sigma_{\Sigma_j,\Delta_j}(\omega_j)$ holds for $i, j \in \{1, 2, 3\}$. Consequently,

$$\sigma_{\Sigma_{\mathsf{bfp}},\Delta_{\mathsf{bfp}}}(\Omega(\Sigma_{\mathsf{bfp}})) = (\sigma_{\Sigma_i,\Delta_i}(\omega_i))^3,$$

where i is any index from $i = 1, 2, 3$. The conditional structures $\sigma_{\Sigma_i,\Delta_i}(\omega_i)$ for $\omega_i \in \Omega(\Sigma_i)$ are shown in Table 3.2 where we use the notation

$$\begin{aligned}
\delta_1 &= (\mathsf{Flies}(c_i)|\mathsf{Bird}(c_i)),\\
\delta_2 &= (\mathsf{Penguin}(c_i) \Rightarrow \mathsf{Bird}(c_i)|\top),\\
\delta_3 &= (\neg\mathsf{Flies}(c_i)|\mathsf{Penguin}(c_i)).
\end{aligned}$$

Consequently, we have

$$\begin{aligned}
\sigma_{\Sigma_{\mathsf{bfp}},\Delta_{\mathsf{bfp}}}(\Omega(\Sigma_{\mathsf{bfp}})) = (&\mathsf{a}_1^+\ \mathsf{a}_2^+\ \mathsf{a}_3^- + \mathsf{a}_1^+\ \mathsf{a}_2^+ + \mathsf{a}_1^-\ \mathsf{a}_2^+\ \mathsf{a}_3^+ + \mathsf{a}_1^-\ \mathsf{a}_2^+\\
&+ \mathsf{a}_2^-\ \mathsf{a}_3^- + \mathsf{a}_2^-\ \mathsf{a}_3^+ + 2 \cdot \mathsf{a}_2^+)^3. \qquad (3.6)
\end{aligned}$$

Factoring this expression out results in a sum of all conditional structures of the possible worlds in $\Omega(\Sigma_{\mathsf{bfp}})$. When consolidating equal conditional structures, the prefactors of the resulting terms are the frequencies with which the respective conditional structures can be observed. By way of a hint, we have

$$\begin{aligned}
\sigma_{\Sigma_{\mathsf{bfp}},\Delta_{\mathsf{bfp}}}(\Omega(\Sigma_{\mathsf{bfp}})) = \quad & (\mathsf{a}_1^+)^3(\mathsf{a}_2^+)^3(\mathsf{a}_3^-)^3\\
&+ 3 \cdot (\mathsf{a}_1^+)^3(\mathsf{a}_2^+)^3(\mathsf{a}_3^-)^2\\
&+ 3 \cdot (\mathsf{a}_1^+)^3(\mathsf{a}_2^+)^3(\mathsf{a}_3^-) \qquad (3.7)\\
&+ \quad (\mathsf{a}_1^+)^3(\mathsf{a}_2^+)^3\\
&+ 6 \cdot (\mathsf{a}_1^+)^2(\mathsf{a}_1^-)(\mathsf{a}_2^+)^3(\mathsf{a}_3^+)(\mathsf{a}_3^-)\\
&+ \dots
\end{aligned}$$

For instance, there is one possible world in which the first two conditionals are verified three times and the third conditional is falsified three times, there are three possible worlds in which the first two conditionals are verified three times and the third conditional is falsified twice, and so on.

Note that we have fixed the domain size $k = |\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}|$ in Example 3.2.4 to $k = 3$, but the arguments that we have made when simplifying the expression $\sigma_{\Sigma_{\mathsf{bfp}},\Delta_{\mathsf{bfp}}}(\Omega(\Sigma_{\mathsf{bfp}}))$ apply to any finite domain size $k \in \mathbb{N}$ as well. Hence, our result can be generalized to arbitrary domain sizes. Actually, the essential concept that we made use of in Example 3.2.4 was the fact that the ground instances of the conditionals in Δ_{bfp} are built on disjoint sets of ground atoms. In a more general setting, this concept is known as the principle of *syntax splitting* [Parikh, 1999; Kern-Isberner and Brewka, 2017; Wilhelm et al., 2017b]. Roughly speaking, if expressions split into syntactically independent parts, then syntax splitting states that reasoning can be performed locally on these parts. If syntax splitting applies to conditionals, then conditional structures factorize which is an essential tool for obtaining a high degree of compression. On a more subtle, formula-based level, syntax splitting corresponds to *decomposable conjunctions*, i.e., conjunctions the conjuncts of which do not share any ground atoms. We will discuss this point in more detail in Section 6.1.

Another strategy which is mentioned in Example 3.2.4 is identifying *isomorphic instances* of conditionals, i.e., instances $\delta_1, \delta_2 \in \mathsf{Inst}_{\Sigma}(\delta)$ of a conditional δ with $\sigma_{\Sigma_1,\delta_1}(\Omega_1(\Sigma_1)) = \sigma_{\Sigma_2,\delta_2}(\Omega_2(\Sigma_2))$ where $\Omega_1(\Sigma_1)$ and $\Omega_2(\Sigma_2)$ are appropriate sets of partial possible worlds. Isomorphic instances allow one to compute conditional structures with respect to one instance and transfer the result to the other instances. Typically, one observes this behavior when constants refer to indistinguishable individuals so that instances are basically the same up to a renaming of the constants. Such interchangeabilities typically lead to taking the conditional structure of one instance to the exponent k when k is the number of isomorphic instances. Both concepts, syntax splitting and considering isomorphic instances, can be utilized if the conditionals are built of formulas which are Boolean combinations of unary predicates.

Proposition 3.2.5: Conditional Structures and Unary Predicates
(cf. Wilhelm et al. [2017b])

Let $\Sigma = (\mathsf{Const}_{\Sigma}, \mathsf{Pred}_{\Sigma})$ be a finite signature with $|\mathsf{Const}_{\Sigma}| \geq 1$, and let

$$\Delta = \{(\psi_1(X)|\phi_1(X)), \ldots, (\psi_n(X)|\phi_n(X))\}$$

be a finite set of conditionals which are built of Boolean combinations of unary predicates, i.e., $\phi_i(X)$ and $\psi_i(X)$ for $i = 1, \ldots, n$ are either of the form $A(X)$ where $A/1$ is a unary predicate from Pred_{Σ} and X is a variable, or they are recursively defined by $\phi'(X) \wedge \psi'(X)$, $\phi'(X) \vee \psi'(X)$, or $\neg\phi'(X)$ where $\phi'(X)$ and $\psi'(X)$ are Boolean combinations of unary predicates within $\mathcal{RL}(\Sigma)$. Further, let

$$\Delta_c = \{(\psi_1(c)|\phi_1(c)), \dots, (\psi_n(c)|\phi_n(c))\}$$

for $c \in \mathsf{Const}_\Sigma$. Then, with an arbitrary $c \in \mathsf{Const}_\Sigma$, we have

$$\sigma_{\Sigma,\Delta}(\Omega(\Sigma)) = \sigma_{\Sigma_c,\Delta_c}(\Omega(\Sigma_c))^{|\mathsf{Const}_\Sigma|} \cdot 2^{\sum_{P \in \mathsf{Pred}_\Sigma \setminus \mathcal{U}_\Delta} |\mathsf{Const}_\Sigma|^{\mathsf{arity}(P)}},$$

where $\Sigma_c = (\{c\}, \mathcal{U}_\Delta)$ is the subsignature of Σ naming the constant c only and where $\mathcal{U}_\Delta$ is the set of unary predicates occurring in Δ.

Proof. Let $\mathsf{Const}_\Sigma = \{c_1, \dots, c_k\}$ with $k \geq 1$. We abbreviate $\Sigma_i = \Sigma_{c_i}$ and $\Delta_i = \Delta_{c_i}$ for $i = 1, \dots, k$. Further, let $\Sigma' = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma \setminus \mathcal{U}_\Delta)$. In the same way as in Example 3.2.4, we compute

$$\sigma_{\Sigma,\Delta}(\Omega(\Sigma)) = \sum_{\omega \in \Omega(\Sigma)} \sigma_{\Sigma,\Delta}(\omega) = \sum_{\omega \in \Omega(\Sigma)} \prod_{c_i \in \mathsf{Const}_\Sigma} \sigma_{\Sigma,\Delta_i}(\omega).$$

Because the conditionals in Δ_i make use of atoms from $\mathsf{grAtom}(\Sigma_i)$ only, and

$$\Omega(\Sigma) = \{\omega' \wedge \bigwedge_{i=1}^{k} \omega_i \mid \omega' \in \Omega(\Sigma'), \qquad \omega_i \in \Omega(\Sigma_i), \quad i = 1, \dots, k\},^{1}$$

we further have

$$\begin{aligned}\sigma_{\Sigma,\Delta}(\Omega(\Sigma)) &= \sum_{\omega' \in \Omega(\Sigma')} \sum_{\omega_1 \in \Omega(\Sigma_1)} \cdots \sum_{\omega_k \in \Omega(\Sigma_k)} \prod_{i=1}^{k} \sigma_{\Sigma_i,\Delta_i}(\omega_i) \\ &= \left(\sum_{\omega' \in \Omega(\Sigma')} 1 \right) \cdot \prod_{i=1}^{k} \sum_{\omega_i \in \Omega(\Sigma_i)} \sigma_{\Sigma_i,\Delta_i}(\omega_i).\end{aligned}$$

Because the sets of conditionals Δ_i for $i = 1, \dots, n$ are all the same up to the names of the constants, i.e., Δ_i can be obtained from Δ_j by replacing every occurrence of c_j in Δ_j by c_i which holds for all $i, j \in \{1, \dots, k\}$,

$$\prod_{i=1}^{k} \sum_{\omega_i \in \Omega(\Sigma_i)} \sigma_{\Sigma_i,\Delta_i}(\omega_i) = (\sigma_{\Sigma_j,\Delta_j}(\omega_i))^k$$

follows where $j \in \{1, \dots, k\}$ is arbitrarily chosen. Further, the number of possible worlds in $\Omega(\Sigma')$ is, according to simple combinatorial arguments,

$$|\Omega(\Sigma')| = 2^{\sum_{P \in \mathsf{Pred}_\Sigma \setminus \mathcal{U}_\Delta} |\mathsf{Const}_\Sigma|^{\mathsf{arity}(P)}}.$$

---

[1] Possibly, up to a permutation of literals in the complete conjunction.

Altogether, we have

$$\sigma_{\Sigma,\Delta}(\Omega(\Sigma)) = \sigma_{\Sigma_c,\Delta_c}(\Omega(\Sigma_c))^{|\mathsf{Const}_\Sigma|} \cdot 2^{\sum_{P \in \mathsf{Pred}_\Sigma \setminus \mathcal{U}_\Delta} |\mathsf{Const}_\Sigma|^{\mathsf{arity}(P)}}$$

for all $c \in \mathsf{Const}_\Sigma$. $\square$

Such compact representations of conditional structures $\sigma_{\Sigma,\Delta}(\Omega(\Sigma))$ as derived in Example 3.2.5 require sophisticated algorithms for their further processing in order to not lose the computational advantages gained through the compact representation. With *condensed iterative scaling*, we will discuss such an algorithm in Section 5.2. The computation of compact representations of $\sigma_{\Sigma,\Delta}(\Omega(\Sigma))$ for more difficult problem classes is thematized in Section 6.4.

Besides syntax splitting and exploiting indistinguishabilities between constants, a third strategy which is implicitly used in Example 3.2.4 is aggregating possible worlds to equivalence classes (cf. Equation (3.7)). We dedicate the remainder of this section to that topic.

▶ Conditional Equivalence of Possible Worlds

The conditional structures of possible worlds induce an equivalence relation on the set of possible worlds $\Omega(\Sigma)$ (cf. Definition A.1.5 in the appendix for a definition of equivalence relations). In this subsection, we briefly discuss this equivalence relation.

Definition 3.2.6: Conditional Equivalence of Possible Worlds

[Kern-Isberner, 2001a]

Let Σ be a finite signature, and let $\Delta \subseteq \mathcal{RCL}(\Sigma)$ be a finite set of conditionals. We define the equivalence relation $\sim_\Delta$ on the set of possible worlds $\Omega(\Sigma)$ by

$$\omega \sim_\Delta \omega' \quad \text{iff} \quad \sigma_{\Sigma,\Delta}(\omega) = \sigma_{\Sigma,\Delta}(\omega'), \quad \text{for} \quad \omega, \omega' \in \Omega(\Sigma).$$

If $\omega \sim_\Delta \omega'$ holds, then we say that ω and ω' are *conditionally equivalent with respect to* Δ or *conditionally equivalent* for short. We denote the equivalence classes induced by $\sim_\Delta$ with $[\omega]_\Delta$ for $\omega \in \Omega(\Sigma)$, and the set of all these equivalence classes with $\Omega_{\sim_\Delta}(\Sigma) = \{[\omega]_\Delta \mid \omega \in \Omega(\Sigma)\}$.

We consider the following obvious characterization of conditional equivalence.

Proposition 3.2.7: Characterization of Conditional Equivalence
Let Σ be a finite signature, let $\Delta \subseteq \mathcal{RCL}(\Sigma)$ be a finite set of conditionals, and let $\omega, \omega' \in \Omega(\Sigma)$ be possible worlds. Then, $\omega \sim_\Delta \omega'$ iff

$$\mathsf{ver}_{\Sigma,\delta}(\omega) = \mathsf{ver}_{\Sigma,\delta}(\omega') \quad \text{and} \quad \mathsf{fal}_{\Sigma,\delta}(\omega) = \mathsf{fal}_{\Sigma,\delta}(\omega') \quad \text{for all} \quad \delta \in \Delta.$$

Proof. The proposition is a direct consequence of the fact that $\sigma_{\Sigma,\Delta}(\omega) = \sigma_{\Sigma,\Delta}(\omega')$ holds if and only if both $\mathsf{ver}_{\Sigma,\delta}(\omega) = \mathsf{ver}_{\Sigma,\delta}(\omega')$ and $\mathsf{fal}_{\Sigma,\delta}(\omega) = \mathsf{fal}_{\Sigma,\delta}(\omega')$ hold for $\delta \in \Delta$. Recall that the generators of conditional structures, a_i^+ and a_i^- for $i = 1, \dots, n$, cannot cancel each other out due to the definition of the free Abelian group $\mathcal{G}(\Delta)$. □

Conditional structures abstract from the syntactical representation of the conditionals in Δ in the sense that they depend on the evaluation of the premises and conclusions of the conditionals only. On the semantical level, possible worlds with the same conditional structure cannot be distinguished based on Δ as they evaluate all conditionals in Δ the same. Hence, possible worlds with the same conditional structure should affect the models of Δ in equal measure regardless of how models of Δ are exactly defined. This principle is called *conditional indifference* [Kern-Isberner and Thimm, 2012] and should be kept in mind when we proceed to probabilistic extensions of qualitative conditionals: It obviously makes no sense to assign different probabilities to conditionally equivalent possible worlds when establishing probabilistic models of (sets of) probabilistic conditionals.

We seize on Example 3.2.4 in order to illustrate the concept of conditional equivalence.

Example 3.2.8
We take Example 3.2.4 as a starting point and consider the signature $\Sigma_{\mathsf{bfp}} = (\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}, \mathsf{Pred}_{\Sigma_{\mathsf{bfp}}})$ with

$$\mathsf{Pred}_{\Sigma_{\mathsf{bfp}}} = \{\mathsf{Bird}/1,\ \mathsf{Flies}/1,\ \mathsf{Penguin}/1\}$$

and domain size $k = |\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}|$ with $k \geq 1$, but we assume that we have already investigated the partial possible worlds in $\Omega(\Sigma_i)$ and are interested in the instantiated set of conditionals

$$\Delta_i = \{(\mathsf{Flies}(c_i)|\mathsf{Bird}(c_i)), (\mathsf{Penguin}(c_i) \Rightarrow \mathsf{Bird}(c_i)|\top), (\neg\mathsf{Flies}(c_i)|\mathsf{Penguin}(c_i))\}$$

with respect to a constant $c_i \in \mathsf{Const}_{\Sigma_{\mathsf{bfp}}}$. We can observe that no two partial possible worlds $\omega, \omega' \in \Omega(\Sigma_i)$ with $\omega \neq \omega'$ are conditionally equivalent with respect to Σ_i and Δ_i except for $\omega_1 = \overline{B(c_i)}F(c_i)\overline{P(c_i)}$ and $\omega_2 = \overline{B(c_i)}\overline{F(c_i)}\overline{P(c_i)}$

(cf. Table 3.2). For these two partial possible worlds we have $\omega_1 \sim_{\Delta_i} \omega_2$ because of $\sigma_{\Delta_i}(\omega_1) = \mathsf{a}_2^+ = \sigma_{\Delta_i}(\omega_2)$. This fact is considered in (3.6): The conditional structures $\sigma_{\Sigma_i,\Delta_i}(\omega_1)$ and $\sigma_{\Sigma_i,\Delta_i}(\omega_2)$ are combined to the term $2 \cdot \mathsf{a}_2^+$ which replaces the term $\mathsf{a}_2^+ + \mathsf{a}_2^+$. We see from this example that the conditional equivalence of possible worlds can also lead to factorizations of conditional structures.

Equivalence classes of possible worlds and their impact on conditional structures have already been discussed in literature (cf. e.g., [Finthammer and Beierle, 2012; Wilhelm et al., 2016]). What is new in this thesis is that we will combine the benefits of conditional equivalence with the other techniques illustrated in Example 3.2.4 and apply them simultaneously to sets of (partial) possible worlds working in the semiring $\mathcal{S}(\Delta)$ on an abstract, algebraic level.

In Section 6.2, we will integrate all the presented techniques into a formal framework for calculating conditional structures which is called *typed model counting*. To anticipate the idea of typed model counting, we will translate sets of conditionals into so-called *structured formulas* gaining the possibility to evaluate several conditionals at the same time with the goal to find compact representations of the conditionals from which the conditional structures can be read easily.

3.3 Relational Probabilistic Conditionals

Relational probabilistic conditionals extend qualitative conditionals by probabilities and, therewith, allow us to quantify the uncertainty expressed by the conditional. Moreover, *open (probabilistic) conditionals* which mention free variables can be used to relate instances of conditionals via sophisticated semantics in various ways. After a short motivation, we discuss the syntax of probabilistic conditionals and the semantics of *closed conditionals* that are built upon sentences. Semantics of *open conditionals*, in particular the *aggregating semantics*, are discussed in Section 3.4.

▶ Motivation of Relational Probabilistic Conditionals

With qualitative conditionals $(\psi|\phi) \in \mathcal{RCL}(\Sigma)$ it is possible to formulate defeasible statements/beliefs (cf. Section 3.1). *Preferential semantics* [Kraus et al., 1990] of qualitative conditionals rate the plausibility of possible worlds, for instance with respect to their conditional structure, and establish a belief state which is usually a hierarchical ordering of the possible worlds. Based on this belief state, plausible inferences can be drawn. Roughly said, possible worlds which verify conditionals are understood to be more plausible than possible worlds which falsify conditionals. This treatment of plausibility is purely comparative. The plausibility of a possible world can be compared to the plausibility of another possible world but there is no absolute degree of how plausible a possible world is and, furthermore, no measure of how much more plausible a possible world is compared to another possible world.

A first step towards a *degree of belief* is made by so-called *ranking functions* or *ordinal conditional functions (*OCF*s)* [Spohn, 1988; Kern-Isberner, 2004] which assign a plausibility rank to possible worlds, usually in form of a natural number. The higher its plausibility rank, the less plausible a possible world is, so that possible worlds with rank zero are the most plausible ones. By taking the distance between ranks into account, we gain a semi-quantitative view on conditionals. For a ranking-based semantics of relational qualitative conditionals see [Kern-Isberner and Thimm, 2012].

Nevertheless, qualitative conditionals are not fully capable of *quantifying* uncertainty like it is present in the statement that "99 % of all birds are able to fly," for instance. Consequently, there have emerged many approaches on incorporating quantification into uncertain reasoning such as *fuzzy logic* [Zadeh, 1965; Novák et al., 1999], *Dempster-Shafter theory* [Dempster, 1967; Shafer, 1976], and *possibility theory* [Dubois and Prade, 1988; Dubois et al., 1991]. Undoubtedly the most prominent methodology for quantification in uncertain reasoning is *probability theory* [Pearl, 1988; Jaynes, 2003; Halpern, 2005; Russell and Norvig, 2010], though. Accordingly, in *Bayesian statistics* [Lee, 2012; MacKay, 2003] degrees of belief are formalized by probabilities in line with the idea of quantifying the validity of a conditional by a *degree of uncertainty* from [Calabrese, 1991].

The popularity of probability theory in the research field of uncertain reasoning is not only because of its long tradition in mathematics [Kolmogoroff, 1933; de Laplace, 1820] but also due to its outstanding properties with respect to *common sense reasoning* which particularly holds for probabilistic models following the principle of maximum entropy [Paris, 1998]. This has already been stated by Pierre-Simon Laplace in his *Théorie Analytique des Probabilités* [de Laplace, 1820; Dale and Laplace, 1995]:

> "On voit, par cet Essai, que la théorie des probabilités n'est, au fond, que le bon sens réduit au calcul. [The theory of probabilities is at bottom nothing but common sense reduced to calculus.]"

Following this tradition, we use a probabilistic framework to express uncertainty in this thesis, too. More precisely, we extend the relational logics $\mathcal{RL}(\Sigma)$ by probabilistic conditionals $(\psi|\phi)[p]$ which formalize statements of the form:

"If ϕ holds, then ψ holds with probability p."

Besides the possibility to quantify uncertainty, our general definition of probabilistic conditionals involves a further enhancement compared to the representation of knowledge by classical sentences. While in sentences all logical variables are bounded by a quantifier, the formulas ϕ and ψ in a probabilistic conditional $(\psi|\phi)[p]$

are allowed to mention free variables. Such conditionals with free variables are called *open*.

Open conditionals leave room for various semantics. Roughly speaking, the probability of an open conditional need not necessarily apply to all individuals which are addressed by the conditional in the same way, but it is possible to formulate semantics that refer to the instances of the conditional collectively. The probability can then be balanced out over all individuals which leads to a higher degree of freedom in the interpretation of conditionals.

For example, the conditional $(\mathsf{Flies}(X)|\mathsf{Bird}(X))[0.99]$ does not necessarily state that for every single bird the probability that this bird is able to fly is assumed to be $p = 0.99$, but the conditional rather states that a prototypical bird is able to fly with this probability. For individual birds the probability of being able to fly may differ from p then. For example, adding the probabilistic sentences $\mathsf{Bird}(\mathsf{tweety})[1]$ and $\mathsf{Flies}(\mathsf{tweety})[0.01]$ which state that Tweety is a bird that is most likely not able to fly does not necessarily contradict $(\mathsf{Flies}(X)|\mathsf{Bird}(X))[0.99]$ as long as there are "enough" volant birds which support the latter conditional. What "enough" exactly means has to be countered by the concrete semantics of open conditionals. These semantics—for logical reasons—generalize the concept of conditional probabilities. Recall that we will discuss such sophisticated semantics in Section 3.4, in particular the *aggregating semantics* [Kern-Isberner and Thimm, 2010; Thimm and Kern-Isberner, 2012] which combines aspects of statistical and subjective probabilities.

▶ Syntax of Relational Probabilistic Conditionals

With *probabilistic conditionals* we define our basic notion that we use to express quantified defeasible statements/beliefs in a probabilistic manner.

Definition 3.3.1: (Relational) Probabilistic Conditional
(cf. e.g., [Kern-Isberner and Thimm, 2010])

Let Σ be a finite signature, let $\delta \in \mathcal{RCL}(\Sigma)$ be a qualitative conditional, and let $p \in [0, 1]$ be a probability. Then, $r = \delta[p]$ is called a *(relational) probabilistic conditional*, where δ is the *logical part* of r and p the *probability* of r. With $\mathcal{RPCL}(\Sigma)$ we denote the set of all probabilistic conditionals over Σ. If $p \in \{0, 1\}$, then r is called a *factual conditional*. Otherwise, if $p \in (0, 1)$, r is a *non-factual conditional* or *real conditional*.

Probabilistic conditionals $r = \delta[p]$ with a probability $p \in \{0, 1\}$ deserve the attribute *factual* because they state that either the conclusion of δ follows from the premise of δ *for sure* (in case of $p = 1$) or it *does not follow at all* (in case of $p = 0$).

Like qualitative conditionals, probabilistic conditionals can be *closed* or *open*. A probabilistic conditional $r = \delta[p]$ is *closed* if δ is closed and it is *open* if δ is open.

Probabilistic conditionals can be instantiated in the same way as qualitative conditionals, and we denote the set of all proper instances of a probabilistic conditional $r \in \mathcal{RPCL}(\Sigma)$ with $\mathsf{Inst}_\Sigma(r)$ in analogy to the set of instances of qualitative conditionals. Note that we annotate the probability value of r to its instances as well. Also the concept of conditional structures and the related notions, in particular the counting functions $\mathsf{ver}_{\Sigma,\delta}$, $\mathsf{fal}_{\Sigma,\delta}$, $\mathsf{app}_{\Sigma,\delta}$, and $\mathsf{napp}_{\Sigma,\delta}$, carry over to probabilistic conditionals in the obvious way. For example, we define $\sigma_{\Sigma,r}(\omega) = \sigma_{\Sigma,\delta}(\omega)$ for the conditional structure of $\omega \in \Omega(\Sigma)$ with respect to $r = \delta[p]$. This is possible because the instances of the probabilistic conditional r are exactly the same as the instances of its logical part δ aside from the annotated probability p.

Example 3.3.2
We consider a signature $\Sigma_{\mathsf{bfp}} = (\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}, \mathsf{Pred}_{\Sigma_{\mathsf{bfp}}})$ with $\mathsf{tweety} \in \mathsf{Const}_{\Sigma_{\mathsf{bfp}}}$ and

$$\mathsf{Pred}_{\Sigma_{\mathsf{bfp}}} = \{\mathsf{Bird}/1,\ \mathsf{Flies}/1,\ \mathsf{Penguin}/1\}.$$

The probabilistic conditionals

$$\begin{aligned} r_1 &= (\mathsf{Flies}(X)|\mathsf{Bird}(X))[0.9], & r_3 &= (\neg\mathsf{Flies}(X)|\mathsf{Penguin}(X))[0.99],\\ r_2 &= (\mathsf{Bird}(X)|\mathsf{Penguin}(X))[1], & r_4 &= (\mathsf{Penguin}(\mathsf{tweety})|\top)[1], \end{aligned}$$

state that birds are usually able to fly, namely with a probability of $p = 0.9$ (r_1), that penguins are birds (r_2) which are most likely not able to fly, namely with probability $p = 0.99$ (r_3), and that Tweety is a penguin (r_4). While the probabilistic conditionals r_2 and r_4 are factual, r_1 and r_3 are non-factual conditionals. The statement in conditional r_3, namely that penguins are not able to fly, is rated as more probable than the statement that birds are able to fly in r_1. This is expressed by the probability of r_3 which is higher than the probability of r_1. Conditional r_4 is closed, the remaining conditionals are open.

Closed probabilistic conditionals admit the intuitive semantics based on the concept of conditional probabilities. We will formally define the semantics of closed probabilistic conditionals first before we consider arbitrary (possibly open) probabilistic conditionals in the next section.

▶ Semantics of Closed Probabilistic Conditionals

The formal semantics of probabilistic conditionals is based on probability distributions over possible worlds (cf. [Halpern, 2010]).

Definition 3.3.3: Probability Distribution Over Possible Worlds

(cf. e.g., [Halpern, 2010])

Let Σ be a finite signature. A function $\mathcal{P} : 2^{\Omega(\Sigma)} \to [0,1]$ is a *probability distribution over the set of possible worlds* $\Omega(\Sigma)$ iff $\mathcal{P}$ satisfies the axioms of Kolmogoroff [Kolmogoroff, 1933], i.e., iff

- $\mathcal{P}(\{\omega\}) \in [0,1]$ for all $\omega \in \Omega(\Sigma)$,
- $\mathcal{P}(\Omega(\Sigma)) = 1$,
- $\mathcal{P}(\mathcal{W}) = \sum_{\omega \in \mathcal{W}} \mathcal{P}(\omega)$ for all $\mathcal{W} \subseteq \Omega(\Sigma)$.

A probability distribution $\mathcal{P} \colon 2^{\Omega(\Sigma)} \to [0,1]$ is fully determined by the elementary probabilities $\mathcal{P}(\{\omega\})$ for $\omega \in \Omega(\Sigma)$. Instead of $\mathcal{P}(\{\omega\})$, we usually write $\mathcal{P}(\omega)$. More generally, we omit the set braces of any set if the set is the only argument of a mapping and there is no risk of confusion. Probability distributions over possible worlds are extended to sentences $\phi \in \mathcal{RL}^S(\Sigma)$ via

$$\mathcal{P}(\phi) = \mathcal{P}(\mathsf{Mod}_\Sigma(\phi)) = \sum_{\omega \in \Omega(\Sigma)\colon\, \omega \models_\Sigma \phi} \mathcal{P}(\omega).$$

Further, $\mathcal{P}$ is a *model* of a *closed* probabilistic conditional $(\psi|\phi)[p]$, iff p equals the *conditional probability* $\mathcal{P}(\psi|\phi)$.

Definition 3.3.4: Semantics of Closed Probabilistic Conditionals

(cf. e.g., [Kern-Isberner and Thimm, 2010])

Let Σ be a finite signature, let $r = (\psi|\phi)[p]$ be a closed probabilistic conditional from $\mathcal{RPCL}(\Sigma)$ and let $\mathcal{P} \colon \Omega(\Sigma) \to [0,1]$ be a probability distribution. Then, $\mathcal{P}$ is a *probabilistic model* of r, written

$$\mathcal{P} \models (\psi|\phi)[p], \quad \text{iff} \quad \mathcal{P}(\phi) > 0 \quad \text{and} \quad \frac{\mathcal{P}(\phi \wedge \psi)}{\mathcal{P}(\phi)} = p.$$

Interpreting closed probabilistic conditionals by conditional probabilities is straightforward and leads to a generally accepted semantics of closed probabilistic conditionals (cf. [Cox, 1946]).

Example 3.3.5
We consider the signature (cf. Example 2.2.2(a))

$$\Sigma_{\mathsf{bow}} = (\emptyset, \{\mathsf{Rain}/0,\ \mathsf{Rainbow}/0,\ \mathsf{Sunshine}/0\})$$

and the closed conditional

$$r_{\mathsf{bow}} = (\mathsf{Rainbow}|\mathsf{Sunshine} \wedge \mathsf{Rain})[0.6].$$

> "If sunshine and rain concur, then a rainbow occurs with a probability of $p = 0.6$."

For a probability distribution $\mathcal{P}$ over $\Omega(\Sigma_{\mathsf{bow}})$ it holds that $\mathcal{P}$ is a probabilistic model of r_{bow}, i.e., $\mathcal{P} \models r_{\mathsf{bow}}$, iff

$$\begin{aligned} \frac{\mathcal{P}(BRS)}{\mathcal{P}(RS)} = \frac{\mathcal{P}(BRS)}{\mathcal{P}(BRS) + \mathcal{P}(\overline{B}RS)} &= 0.6 \\ \text{and} \qquad \mathcal{P}(BRS) + \mathcal{P}(\overline{B}RS) &> 0, \end{aligned} \tag{3.8}$$

where we abbreviate $B = \mathsf{Rainbow}$, $S = \mathsf{Sunshine}$, and $R = \mathsf{Rain}$. For example, the probability distribution $\mathcal{P}$ with

$$\begin{aligned} &\mathcal{P}(BRS) = 0.6, \qquad \mathcal{P}(\overline{B}RS) = 0.4, \\ \text{and} \quad &\mathcal{P}(\omega) = 0 \quad \text{for all} \quad \omega \in \Omega(\Sigma_{\mathsf{bow}}) \setminus \{BRS, \overline{B}RS\} \end{aligned}$$

is a probabilistic model of r_{bow} because in this case (3.8) becomes

$$\frac{0.6}{0.6 + 0.4} = 0.6 \quad \text{and} \quad 0.6 + 0.4 = 1 > 0,$$

which is obviously true. Note that (3.8) is the standard definition of the conditional probability $\mathcal{P}(B|RS) = 0.6$.

We define *probabilistic sentences* $\phi[p]$ as pairs of sentences $\phi \in \mathcal{RL}^S(\Sigma)$ and probabilities $p \in [0, 1]$ so that

$$\mathcal{P} \models_\Sigma \phi[p] \quad \text{iff} \quad \mathcal{P} \models_\Sigma (\phi|\top)[p],$$

which holds iff $\mathcal{P}(\phi) = p$. Note that this formalization of probabilistic sentences subsumes classical logic in the sense that $\mathcal{P} \models_\Sigma \phi$ holds iff $\mathcal{P}(\omega) = 0$ for all $\omega \notin \mathsf{Mod}_\Sigma(\phi)$ when identifying $\mathcal{P} \models_\Sigma \phi$ with $\mathcal{P} \models_\Sigma \phi[1]$ for sentences $\phi \in \mathcal{RL}^S(\Sigma)$. This is in accordance with the general intent of unifying probability theory and deductive logic [Nilsson, 1986].

While the semantics of closed probabilistic conditionals is very straightforward and out of question, the semantics of *open* probabilistic conditionals is not so obvious because there is no clear interpretation of open formulas with probabilities. The logical part of an open conditional cannot be interpreted with respect to possible worlds without instantiating the conditional. At least there is no generally accepted semantics for that and, hence, there is a need to consider a more sophisticated semantics of open probabilistic conditionals which takes the instances of the conditional into account. In particular, open probabilistic conditionals demand a more sophisticated interpretation than the concept of conditional probabilities provides. In the next section, we discuss with the *aggregating semantics* [Kern-Isberner and Thimm, 2010; Thimm and Kern-Isberner, 2012] such a sophisticated semantics of open probabilistic conditionals.

3.4 Semantics of Open Probabilistic Conditionals

In this thesis, we will rely on the so-called *aggregating semantics* [Kern-Isberner and Thimm, 2010; Thimm and Kern-Isberner, 2012] when interpreting open probabilistic conditionals. The idea behind the aggregating semantics is to mimic statistical probabilities from a subjective point of view. The probabilities are assigned to possible worlds in such a way that it is possible to make probabilistic conditional statements about the general behavior of groups of individuals as well as about single individuals. The probabilities assigned to statements about individuals are balanced out such that they fit the probabilities assigned to the statements about the groups of individuals provided that consistency is achievable. In this way, probabilities under the aggregating semantics can be understood as degrees of belief in accordance with type 2 probabilities in regard to the classification of Halpern [Halpern, 1990].

In this section, we give a formal definition of the *aggregating semantics* as well as a characterization which makes use of so-called *feature functions*. After that, we have a closer look at the feature functions themselves and briefly discuss alternative semantics of open probabilistic conditionals.

▶ Aggregating Semantics

The aggregating semantics extends the concept of conditional probabilities by integrating the counts of possible worlds in which the ground instances of the conditional are verified or applicable. Possible worlds in which several ground instances are verified or applicable are considered multiple times. Therewith, the subjective nature of conditional probabilities where probabilities are understood as degrees of belief is enriched by statistical arguments.

Definition 3.4.1: Aggregating Semantics

[Kern-Isberner and Thimm, 2010]

Let Σ be a finite signature, and let $r \in \mathcal{RPCL}(\Sigma)$ with $r = (\psi|\phi)_{\mathsf{CS}}[p]$ be a (possibly open) probabilistic conditional.[a] A probability distribution $\mathcal{P}\colon 2^{\Omega(\Sigma)} \to [0,1]$ is a *probabilistic model of* r *under the aggregating semantics*, or, here, a *probabilistic model of* r for short, written

$$\mathcal{P} \models r, \quad \text{iff} \quad \textstyle\sum_{(\psi'|\phi')[p] \in \mathsf{Inst}_\Sigma(r)} \mathcal{P}(\phi') > 0 \quad \text{and} \quad \frac{\sum_{(\psi'|\phi')[p] \in \mathsf{Inst}_\Sigma(r)} \mathcal{P}(\phi' \wedge \psi')}{\sum_{(\psi'|\phi')[p] \in \mathsf{Inst}_\Sigma(r)} \mathcal{P}(\phi')} = p. \tag{3.9}$$

We denote the set of all probabilistic models of r by $\mathsf{probMod}_\Sigma(r)$.

---

[a] Recall that CS is a (possibly empty) set of domain constraints (cf. Section 2.3).

The aggregating semantics coincides with conditional probabilities if the aggregating semantics is applied to closed probabilistic conditionals (cf. Definition 3.3.4 which determines the semantics of closed probabilistic conditionals based on conditional probabilities). That is, if $r = (\psi|\phi)[p]$ is a closed probabilistic conditional, (3.9) reduces to

$$\mathcal{P} \models r \quad \text{iff} \quad \mathcal{P}(\phi) > 0 \quad \text{and} \quad \frac{\mathcal{P}(\phi \wedge \psi)}{\mathcal{P}(\phi)} = p.$$

This is the case because r is the only instance of itself then, $\mathsf{Inst}_\Sigma(r) = \{r\}$, and the sums in (3.9) reduce to single summands. This coincidence justifies the use of the symbol $\models$ for the probabilistic satisfaction of conditionals according to the aggregating semantics and we may apply the aggregating semantics to all conditionals in $\mathcal{RPCL}(\Sigma)$ without differentiating between open and closed conditionals.

Example 3.4.2

Let $\Sigma_{\mathsf{bf}} = (\mathsf{Const}_{\Sigma_{\mathsf{bf}}}, \mathsf{Pred}_{\Sigma_{\mathsf{bf}}})$ be the signature given by

$$\mathsf{Const}_{\Sigma_{\mathsf{bf}}} = \{a, b, c\} \quad \text{and} \quad \mathsf{Pred}_{\Sigma_{\mathsf{bf}}} = \{\mathsf{Bird}/1,\ \mathsf{Flies}/1\}.$$

A probability distribution $\mathcal{P}$ over $\Omega(\Sigma_{\mathsf{bf}})$ is a probabilistic model of the open probabilistic conditional

$$r_{\mathsf{bf}} = (\mathsf{Flies}(X)|\mathsf{Bird}(X))[0.9]$$

"Birds usually fly with probability 0.9."

iff

$$\begin{aligned} & & \frac{\mathcal{P}(B(a)F(a)) + \mathcal{P}(B(b)F(b)) + \mathcal{P}(B(c)F(c))}{\mathcal{P}(B(a)) + \mathcal{P}(B(b)) + \mathcal{P}(B(c))} &= 0.9 \\ & \text{and} & \mathcal{P}(B(a)) + \mathcal{P}(B(b)) + \mathcal{P}(B(c)) &> 0, \end{aligned} \tag{3.10}$$

where $B = \mathsf{Bird}$ and $F = \mathsf{Flies}$. Condition (3.10) makes visible how the three instances of the conditional r_{bf}, here one instance per constant from $\mathsf{Const}_{\Sigma_{\mathsf{bf}}}$, are combined by the aggregating semantics. Basically, the conditional probabilities of the single instances of the conditional are plugged together by summing up the numerators and the denominators of the respective conditional probabilities separately and by building the quotient of both sums thereafter.

We now give a characterization of the aggregating semantics based on so-called *feature functions* which makes it more clear what we mean by "the aggregating semantics mimics statistical probabilities from a subjective point of view."

Definition 3.4.3: Feature Function *(cf. e.g., [Fisseler, 2010])*

Let Σ be a finite signature, let $r = \delta[p]$ be a probabilistic conditional in $\mathcal{RPCL}(\Sigma)$, and let $\omega \in \Omega(\Sigma)$ be a possible world. Then,

$$f_{\Sigma,r}(\omega) = \mathsf{ver}_{\Sigma,r}(\omega) - p \cdot \mathsf{app}_{\Sigma,r}(\omega)$$

is called the *feature function* of ω with respect to r.

Feature functions combine the logical and the probabilistic information of conditionals by aggregating the counting functions $\mathsf{ver}_{\Sigma,r}(\omega)$ and $\mathsf{fal}_{\Sigma,r}(\omega)$ (cf. Definition 3.1.11) as well as the probability p into a single real number. It will be shown that feature functions convey exactly the information needed for the evaluation of conditionals based on the aggregating semantics (cf. Proposition 3.4.5). Beforehand, we show that the value of the feature function with respect to an open conditional r applied to a possible world ω is the sum of the feature function values with respect to the instances of r.

Proposition 3.4.4: Characterization of Feature Functions
Let Σ be a finite signature, let $r \in \mathcal{RPCL}(\Sigma)$ be a probabilistic conditional, and let $\omega \in \Omega(\Sigma)$ be a possible world. Then,

$$f_{\Sigma,r}(\omega) = \sum_{r' \in \mathsf{Inst}_\Sigma(r)} f_{\Sigma,r'}(\omega).$$

Proof. Let $r = (\psi|\phi)_{\mathsf{CS}}[p]$ with $\psi, \phi \in \mathcal{RL}(\Sigma)$, and let $p \in [0,1]$. We have

$$\begin{aligned}
f_{\Sigma,r}(\omega) &= \mathsf{ver}_{\Sigma,r}(\omega) - p \cdot \mathsf{app}_{\Sigma,r}(\omega) \\
&= |\{(\psi'|\phi')[p] \in \mathsf{Inst}_\Sigma(r) \mid \omega \models_\Sigma \phi'\psi'\}| - p \cdot |\{(\psi'|\phi')[p] \in \mathsf{Inst}_\Sigma(r) \mid \omega \models_\Sigma \phi'\}| \\
&= (\textstyle\sum_{r' \in \mathsf{Inst}_\Sigma(r)\colon \omega \models_\Sigma \phi'\psi'} 1) - p \cdot (\textstyle\sum_{r' \in \mathsf{Inst}_\Sigma(r)\colon \omega \models_\Sigma \phi'} 1) \\
&= \sum_{r' \in \mathsf{Inst}_\Sigma(r)} \mathsf{ver}_{\Sigma,r'}(\omega) - p \cdot \sum_{r' \in \mathsf{Inst}_\Sigma(r)} \mathsf{app}_{\Sigma,r'}(\omega) \\
&= \sum_{r' \in \mathsf{Inst}_\Sigma(r)} (\mathsf{ver}_{\Sigma,r'}(\omega) - p \cdot \mathsf{app}_{\Sigma,r'}(\omega)) \\
&= \sum_{r' \in \mathsf{Inst}_\Sigma(r)} f_{\Sigma,r'}(\omega),
\end{aligned}$$

which proves the proposition. ☐

Based on feature functions, we can characterize the aggregating semantics as follows.

Proposition 3.4.5: Characterization of the Aggregating Semantics
Let Σ be a finite signature, let $r \in \mathcal{RPCL}(\Sigma)$ be a probabilistic conditional, and let $\mathcal{P}\colon 2^{\Omega(\Sigma)} \to [0,1]$ be a probability distribution. Then,

$$\begin{aligned}
\mathcal{P} \models_\Sigma r \quad \text{iff} \quad & \textstyle\sum_{\omega \in \Omega(\Sigma)} f_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) = 0 \\
\text{and} \quad & \textstyle\sum_{\omega \in \Omega(\Sigma)} \mathsf{app}_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) > 0.
\end{aligned} \tag{3.11}$$

Proof. Let $r = (\psi|\phi)_{\mathsf{CS}}[p]$ with $\psi, \phi \in \mathcal{RL}(\Sigma)$, and let $p \in [0,1]$. Further, let ξ be any Boolean expression which can be either *true* or *false*, and let

$$\mathbb{1}_\xi = \begin{cases} 1 & \text{if} \quad \xi = \mathit{true} \\ 0 & \text{otherwise} \end{cases}$$

be the indicator function for ξ. The condition $\sum_{\omega \in \Omega(\Sigma)} \mathsf{app}_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) > 0$ is equivalent to the condition $\sum_{(\psi'|\phi')[p] \in \mathsf{Inst}_\Sigma(r)} \mathcal{P}(\phi') > 0$ from the definition of the aggregating semantics (cf. Definition 3.4.1) because

$$\begin{aligned}
\sum_{(\psi'|\phi')[p]\in\mathsf{Inst}_\Sigma(r)} \mathcal{P}(\phi') &= \sum_{(\psi'|\phi')[p]\in\mathsf{Inst}_\Sigma(r)} \; \sum_{\omega\in\Omega(\Sigma)\colon\, \omega\models_\Sigma\phi'} \mathcal{P}(\omega) \\
&= \sum_{(\psi'|\phi')[p]\in\mathsf{Inst}_\Sigma(r)} \; \sum_{\omega\in\Omega(\Sigma)} \mathbb{1}_{\omega\models_\Sigma\phi'} \cdot \mathcal{P}(\omega) \\
&= \sum_{\omega\in\Omega(\Sigma)} \mathcal{P}(\omega) \cdot \sum_{(\psi'|\phi')[p]\in\mathsf{Inst}_\Sigma(r)} \mathbb{1}_{\omega\models_\Sigma\phi'} \\
&= \sum_{\omega\in\Omega(\Sigma)} \mathcal{P}(\omega) \cdot \mathsf{app}_{\Sigma,r}(\omega).
\end{aligned}$$

Analogously, provided that $\sum_{\omega\in\Omega(\Sigma)} \mathsf{app}_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) > 0$ holds, one has

$$\begin{aligned}
& \frac{\sum_{(\psi'|\phi')[p]\in\mathsf{Inst}_\Sigma(r)} \mathcal{P}(\phi' \wedge \psi')}{\sum_{(\psi'|\phi')[p]\in\mathsf{Inst}_\Sigma(r)} \mathcal{P}(\phi')} = p \\
\text{iff}\quad & \frac{\sum_{(\psi'|\phi')[p]\in\mathsf{Inst}_\Sigma(r)} \sum_{\omega\in\Omega(\Sigma)\colon\, \omega\models_\Sigma\phi'\psi'} \mathcal{P}(\omega)}{\sum_{(\psi'|\phi')[p]\in\mathsf{Inst}_\Sigma(r)} \sum_{\omega\in\Omega(\Sigma)\colon\, \omega\models_\Sigma\phi'} \mathcal{P}(\omega)} = p \\
\text{iff}\quad & \frac{\sum_{(\psi'|\phi')[p]\in\mathsf{Inst}_\Sigma(r)} \sum_{\omega\in\Omega(\Sigma)} \mathbb{1}_{\omega\models_\Sigma\phi'\psi'} \cdot \mathcal{P}(\omega)}{\sum_{(\psi'|\phi')[p]\in\mathsf{Inst}_{\mathsf{Const}_\Sigma}(r)} \sum_{\omega\in\Omega(\Sigma)} \mathbb{1}_{\omega\models_\Sigma A'} \cdot \mathcal{P}(\omega)} = p \\
\overset{(*)}{\text{iff}}\quad & \frac{\sum_{\omega\in\Omega(\Sigma)} \mathcal{P}(\omega) \cdot \left(\sum_{(\psi'|\phi')[p]\in\mathsf{Inst}_\Sigma(r)} \mathbb{1}_{\omega\models_\Sigma\phi'\psi'}\right)}{\sum_{\omega\in\Omega(\Sigma)} \mathcal{P}(\omega) \cdot \left(\sum_{(\psi'|\phi')[p]\in\mathsf{Inst}_\Sigma(r)} \mathbb{1}_{\omega\models_\Sigma\phi'}\right)} = p \\
\text{iff}\quad & \frac{\sum_{\omega\in\Omega(\Sigma)} \mathcal{P}(\omega) \cdot \mathsf{ver}_{\Sigma,r}(\omega)}{\sum_{\omega\in\Omega(\Sigma)} \mathcal{P}(\omega) \cdot \mathsf{app}_{\Sigma,r}(\omega)} = p \\
\text{iff}\quad & \sum_{\omega\in\Omega(\Sigma)} \mathcal{P}(\omega) \cdot \mathsf{ver}_{\Sigma,r}(\omega) = p \cdot \sum_{\omega\in\Omega(\Sigma)} \mathcal{P}(\omega) \cdot \mathsf{app}_{\Sigma,r}(\omega) \\
\text{iff}\quad & \sum_{\omega\in\Omega(\Sigma)} \mathcal{P}(\omega) \cdot \big(\mathsf{ver}_{\Sigma,r}(\omega) - p \cdot \mathsf{app}_{\Sigma,r}(\omega)\big) = 0 \\
\text{iff}\quad & \sum_{\omega\in\Omega(\Sigma)} f_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) = 0.
\end{aligned}$$

The important step of this proof is the commutation of the sums over the instances of r and the sums over the possible worlds in $\Omega(\Sigma)$ (cf. $(*)$) which makes it possible to turn the aggregating semantics into a counting problem where (appropriate) instances of r have to be counted for each possible world $\omega \in \Omega(\Sigma)$. □

In order to better understand the aggregating semantics, we are ready to analyze some extreme cases now.

▶ Statistical and Subjective View on the Aggregating Semantics

There are two fundamentally different views on probabilities. On the one hand, probabilities can be understood as purely *statistical frequencies* obtained by the repeated execution of a random experiment, e.g., the toss of a fair coin. And, on the other hand, probabilities can be understood as *subjective degrees of a reasoner's beliefs* in a specific outcome of an event. Both views have their own right to exist (cf. e.g., [Bartlett, 1936]) and are smoothly combined in the aggregating semantics. Because probabilities are defined over possible worlds in this thesis, they are principally *subjective probabilities of type 2* according to the classification of [Halpern, 1990]. However, the aggregating semantics grounds this subjective view on probabilities on statistical arguments. This interplay of statistical and subjective aspects of probabilities within the aggregating semantics becomes clear when considering specific extreme cases, namely the cases where the underlying probability distribution is either a *Dirac distribution* or the *uniform distribution.* We discuss these cases in the following.

First, we assume that $\mathcal{P}' \colon \Omega(\Sigma) \to [0,1]$ is a *Dirac distribution* which maps a single possible world $\omega' \in \Omega(\Sigma)$ to the probability $\mathcal{P}'(\omega') = 1$ and all remaining possible worlds $\omega \in \Omega(\Sigma)$ with $\omega \neq \omega'$ to $\mathcal{P}'(\omega) = 0$. Then, $\mathcal{P}'$ formalizes the belief state of a reasoner who is sure that ω' represents the real world, and the aggregating semantics (cf. Definition 3.4.1) becomes

$$\mathcal{P}' \models r \quad \text{iff} \quad \frac{\mathsf{ver}_{\Sigma,r}(\omega')}{\mathsf{app}_{\Sigma,r}(\omega')} = p \quad \text{and} \quad \mathsf{app}_{\Sigma,r}(\omega') > 0. \tag{3.12}$$

This holds because $\sum_{\omega \in \Omega(\Sigma)} f_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) = 0$ (cf. the characterization of the aggregating semantics in Proposition 3.4.5) reduces to $f_{\Sigma,r}(\omega') = 0$ which is equivalent to $\mathsf{ver}_{\Sigma,r}(\omega') - p \cdot \mathsf{app}_{\Sigma,r}(\omega') = 0$ and which can easily be rearranged to (3.12). Hence, in this case the aggregating semantics demands that the relative frequency of the number of verified instances of r in ω' measured against the number of applicable instances equals p. This corresponds to a purely statistical view on probabilities and (3.12) can be validated by simply counting the models of appropriate sentences in $\mathcal{RL}^S(\Sigma)$.

In our second extreme case, let $\mathcal{P}^u_\Sigma$ be the *uniform distribution* on $\Omega(\Sigma)$ which is defined by $\mathcal{P}^u_\Sigma(\omega) = \frac{1}{|\Omega(\Sigma)|}$ for all possible worlds $\omega \in \Omega(\Sigma)$. The uniform distribution reflects that the reasoner has no opinion which possible world describes the real world best. Then, the aggregating semantics becomes

$$\mathcal{P}^u_\Sigma \models r \quad \text{iff} \quad \frac{\sum_{\omega \in \Omega(\Sigma)} \mathsf{ver}_r(\omega)}{\sum_{\omega \in \Omega(\Sigma)} \mathsf{app}_r(\omega)} = p \quad \text{and} \quad \sum_{\omega \in \Omega(\Sigma)} \mathsf{app}_r(\omega) > 0 \tag{3.13}$$

because $\sum_{\omega \in \Omega(\Sigma)} f_r(\omega) \cdot \mathcal{P}^u_\Sigma(\omega) = |\Omega(\Sigma)|^{-1} \cdot \sum_{\omega \in \Omega(\Sigma)} f_r(\omega)$ and multiplying both sides in (3.11) with the factor $|\Omega(\Sigma)|$ yields $\sum_{\omega \in \Omega(\Sigma)} f_r(\omega) = 0$ from which (3.13) follows. That is, again, the aggregating semantics means counting verified and applicable instances of r, here balanced over all possible worlds in $\Omega(\Sigma)$.

Hence, in the extreme cases where $\mathcal{P}$ is maximally informative ($\mathcal{P} = \mathcal{P}'$) respectively maximally uninformative ($\mathcal{P} = \mathcal{P}^u_\Sigma$), the aggregating semantics is a purely statistical semantics. In the middle cases where $\mathcal{P}$ expresses an uncertain opinion of a reasoner, i.e., if $\mathcal{P} \neq \mathcal{P}'$ and $\mathcal{P} \neq \mathcal{P}^u_\Sigma$, the aggregating semantics means weighting the relative frequencies ($\mathsf{ver}_{\Sigma,r}(\omega)$ measured against $\mathsf{app}_{\Sigma,r}(\omega)$ for $\omega \in \Omega(\Sigma)$) with the subjective estimation of the reasoner that ω formalizes the real world. This means that the subjective and the statistical views on probabilities overlap. Technically speaking, the aggregating semantics can be understood as a weighted model counting problem with weights $\mathcal{P}(\omega)$ for $\omega \in \Omega(\Sigma)$.

We now have a closer look at the feature functions which are used in the characterization of the aggregating semantics in Proposition 3.4.5.

▶ A Closer Look at Feature Functions

The feature function $f_{\Sigma,r} \colon \Omega(\Sigma) \to \mathbb{R}$ defined by $f_{\Sigma,r}(\omega) = \mathsf{ver}_{\Sigma,r}(\omega) - p \cdot \mathsf{app}_{\Sigma,r}(\omega)$ (cf. Definition 3.4.3), which occurs in the characterization of the aggregating semantics of a probabilistic conditional $r \in \mathcal{RPCL}(\Sigma)$ in Proposition 3.4.5, maps possible worlds to (possibly negative) real numbers and incorporates both the logical information of the conditional $r = \delta[p]$ in terms of the counts $\mathsf{ver}_{\Sigma,r}(\omega)$ and $\mathsf{app}_{\Sigma,r}(\omega)$, and the probabilistic information of r in terms of the probability value p. In particular, $f_{\Sigma,r}$ abstracts from the syntactical representation of the logical information provided by r and, thus, condenses the information provided by r which is relevant for establishing a model of r (under the aggregating semantics) to a minimum. This means, the actual formulas in the premise and the conclusion of a conditional become superfluous once the feature functions are derived. One can say that feature functions deserve their name because they feature all the information that is necessary to evaluate the conditional with respect to the aggregating semantics. More formally expressed, feature functions constitute *sufficient statistics* (cf. [Fisher, 1922]) for the models of conditionals.

Basically, feature functions differ from conditional structures in the consideration of the probability p which, however, is the same for every instance of the respective conditional. Hence, determining conditional structures is the crucial part for setting up the feature functions and, therewith, the condition (3.11) of the aggregating semantics.

It is easy to see that conditionally equivalent possible worlds yield the same

ω	$\lvert[\omega]_{\sim_{\Sigma_{\mathsf{bf}},\{r_{\mathsf{bf}}\}}}\rvert$	$\mathsf{ver}_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$	$\mathsf{fal}_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$	$f_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$
$B(a)B(b)B(c)F(a)F(b)F(c)$	1	3	0	0.3
$B(a)B(b)B(c)F(a)F(b)\overline{F(c)}$	3	2	1	-0.7
$B(a)B(b)B(c)F(a)\overline{F(b)}\,\overline{F(c)}$	3	1	2	-1.7
$B(a)B(b)B(c)\overline{F(a)}\,\overline{F(b)}\,\overline{F(c)}$	1	0	3	-2.7
$B(a)B(b)\overline{B(c)}F(a)F(b)F(c)$	6	2	0	0.2
$B(a)B(b)\overline{B(c)}F(a)\overline{F(b)}F(c)$	12	1	1	-0.8
$B(a)B(b)\overline{B(c)}\,\overline{F(a)}\,\overline{F(b)}F(c)$	6	0	2	-1.8
$B(a)\overline{B(b)}\,\overline{B(c)}F(a)F(b)F(c)$	12	1	0	0.1
$B(a)\overline{B(b)}\,\overline{B(c)}\,\overline{F(a)}F(b)F(c)$	12	0	1	-0.9
$\overline{B(a)}\,\overline{B(b)}\,\overline{B(c)}F(a)F(b)F(c)$	8	0	0	0

Table 3.3: Counts $\mathsf{ver}_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$ and $\mathsf{fal}_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$ and feature function values $f_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$ of the possible worlds $\omega \in \Omega(\Sigma_{\mathsf{bf}})$ with respect to the conditional $r_{\mathsf{bf}} = (\mathsf{Flies}(X)|\mathsf{Bird}(X))[0.9]$ and $|\mathsf{Const}_{\Sigma_{\mathsf{bf}}}| = 3$. For each equivalence class $[\omega]_{\sim_{\Sigma_{\mathsf{bf}},\{r_{\mathsf{bf}}\}}} \in \Omega_{\sim_{\Sigma_{\mathsf{bf}},\{r_{\mathsf{bf}}\}}}(\Sigma_{\mathsf{bf}})$ one representative is shown.

feature function values, i.e., for $\omega, \omega' \in \Omega(\Sigma)$,

$$\sigma_{\Sigma,r}(\omega) = \sigma_{\Sigma,r}(\omega') \quad \text{implies} \quad f_{\Sigma,r}(\omega) = f_{\Sigma,r}(\omega').$$

On the other hand, possible worlds which are not conditionally equivalent may lead to the same feature function value, too. That is, $f_{\Sigma,r}(\omega) = f_{\Sigma,r}(\omega')$ does not imply $\sigma_{\Sigma,r}(\omega) = \sigma_{\Sigma,r}(\omega')$. For example, all possible worlds $\omega \in \Omega(\Sigma)$ for which the ratio $\frac{\mathsf{fal}_{\Sigma,r}(\omega)}{\mathsf{ver}_{\Sigma,r}(\omega)}$ equals $\frac{1-p}{p}$ have the feature function value $f_{\Sigma,r}(\omega) = 0$ because, in this case,

$$\begin{aligned}
f_{\Sigma,r}(\omega) &= \mathsf{ver}_{\Sigma,r}(\omega) - p \cdot \mathsf{app}_{\Sigma,r}(\omega) \\
&= \mathsf{ver}_{\Sigma,r}(\omega) - p \cdot (\mathsf{ver}_{\Sigma,r}(\omega) + \mathsf{fal}_{\Sigma,r}(\omega)) \\
&= \mathsf{ver}_{\Sigma,r}(\omega) - p \cdot \left(\mathsf{ver}_{\Sigma,r}(\omega) + \frac{1-p}{p} \cdot \mathsf{ver}_{\Sigma,r}(\omega)\right) \\
&= \mathsf{ver}_{\Sigma,r}(\omega) - p \cdot \left(\frac{1}{p} \cdot \mathsf{ver}_{\Sigma,r}(\omega)\right) \\
&= \mathsf{ver}_{\Sigma,r}(\omega) - \mathsf{ver}_{\Sigma,r}(\omega) = 0.
\end{aligned}$$

Consequently, conditional equivalence induces a partition on the set of possible worlds which refines the partition that is induced by feature functions.

Example 3.4.6

We continue Example 3.4.2(b) and consider the signature Σ_{bf} given by

$$\mathsf{Const}_{\Sigma_{\mathsf{bf}}} = \{a, b, c\} \quad \text{and} \quad \mathsf{Pred}_{\Sigma_{\mathsf{bf}}} = \{\mathsf{Bird}/1,\ \mathsf{Flies}/1\}$$

and the conditional

$$r_{\mathsf{bf}} = (\mathsf{Flies}(X)|\mathsf{Bird}(X))[0.9].$$

Table 3.3 shows the counts $\mathsf{ver}_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$ and $\mathsf{fal}_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$ as well as the feature function values $f_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$ of the possible worlds $\omega \in \Omega(\Sigma_{\mathsf{bf}})$. One representative per equivalence class $[\omega]_{\sim_{\Sigma_{\mathsf{bf}},\{r_{\mathsf{bf}}\}}} \in \Omega_{\sim_{\Sigma_{\mathsf{bf}},\{\omega\}_{\mathsf{bf}}}}(\Sigma_{\mathsf{bf}})$ is shown. In this example, the equivalence classes in $\Omega_{\sim_{\Sigma_{\mathsf{bf}},\{r_{\mathsf{bf}}\}}}(\Sigma_{\mathsf{bf}})$ are the same as the equivalence classes which one would obtain when partitioning $\Omega(\Sigma_{\mathsf{bf}})$ according to the feature function values. Intuitively, the chance that different equivalence classes $[\omega]_{\Sigma_{\mathsf{bf}},\{r_{\mathsf{bf}}\}}$ lead to the same feature function value increases when $k = |\mathsf{Const}_{\Sigma_{\mathsf{bf}}}|$ is large, provided that the probability p is a rational number.

In Example 3.4.6 there are 64 possible worlds but only 10 different feature function values. This feeds the hope that the aggregating semantics according to (3.11) can be set up more efficiently by combining summands, as we have done this for conditional structures in Example 3.2.4, in contrast to calculating the summands independently for every possible world. However, this is only possible if the probabilities $\mathcal{P}(\omega)$ in (3.11) behave well, too. More precisely, this means that for possible worlds $\omega, \omega' \in \Omega(\Sigma)$ with the same feature function values, i.e., with $f_{\Sigma,r}(\omega) = f_{\Sigma,r}(\omega')$, the probabilities $\mathcal{P}(\omega)$ and $\mathcal{P}(\omega')$ have to be equal as well. This requirement can be seen as a mitigated variant of the property *conditional indifference* [Kern-Isberner and Thimm, 2012].

Definition 3.4.7: Conditional Indifference

(based on [Kern-Isberner and Thimm, 2012])

Let Σ be a finite signature, let $r \in \mathcal{RPCL}(\Sigma)$ be a probabilistic conditional, and let $\mathcal{P} \colon \Omega(\Sigma) \to [0, 1]$ be a probability distribution. Then, $\mathcal{P}$ is *conditionally indifferent* with respect to r iff $\sigma_{\Sigma,r}(\omega) = \sigma_{\Sigma,r}(\omega')$ implies $\mathcal{P}(\omega) = \mathcal{P}(\omega')$ for $\omega, \omega' \in \Omega(\Sigma)$. Further, $\mathcal{P}$ is conditionally indifferent with respect to a set of conditionals $\mathcal{B} \subseteq \mathcal{RPCL}(\Sigma)$ iff $\mathcal{P}$ is conditionally indifferent with respect to all conditionals in $\mathcal{B}$.

It is not reasonable to assign different probabilities to possible worlds with the same conditional structure when establishing models of probabilistic conditionals because this discrimination is not supported by the conditional itself. Conditional indifference is not satisfied by all models $\mathcal{P}$ of a conditional r, though. Later on, we will see that the *maximum entropy model* does satisfy conditional indifference (cf. Proposition 4.3.5). Actually, the maximum entropy model satisfies the following accentuation of conditional indifference based on feature functions as well.

Definition 3.4.8: Feature Function Indifference
Let Σ be a finite signature, let $r \in \mathcal{RPCL}(\Sigma)$ be a probabilistic conditional, and let $\mathcal{P}\colon \Omega(\Sigma) \to [0,1]$ be a probability distribution. Then, $\mathcal{P}$ is *feature function indifferent* with respect to r iff $f_{\Sigma,r}(\omega) = f_{\Sigma,r}(\omega')$ implies $\mathcal{P}(\omega) = \mathcal{P}(\omega')$ for $\omega, \omega' \in \Omega(\Sigma)$. Further, $\mathcal{P}$ is *feature function indifferent* with respect to a set of conditionals $\mathcal{B} \subseteq \mathcal{RPCL}(\Sigma)$ iff $\mathcal{P}$ is feature function indifferent with respect to all conditionals in $\mathcal{B}$.

Because conditional equivalence implies equal feature function values, feature function indifference implies conditional indifference. But, as discussed above, this does not hold the other way around.

Sometimes it is useful to consider standardized versions of feature functions with a co-domain (= range) that is restricted to a specific compact interval. We discuss standardized feature functions in the next paragraph.

▶ Standardized Feature Functions

We have a closer look at the co-domain of feature functions and introduce so-called *standardized feature functions* which normalize feature functions to $[-1,1]$. This can be useful, for instance, if an a priori estimation of the feature function values is needed, especially when applying numerical methods to find probabilistic models of conditionals. Obviously, the following estimation of feature function values holds.

Proposition 3.4.9: Upper and Lower Bounds of Feature Function
Let Σ be a finite signature, let $r = (B|A)_{\mathsf{CS}}[p]$ be a probabilistic conditional in $\mathcal{RPCL}(\Sigma)$, and let $\omega \in \Omega(\Sigma)$ be a possible world. Then,

$$-p \cdot |\mathsf{Inst}_\Sigma(r)| \leq f_{\Sigma,r}(\omega) \leq (1-p) \cdot |\mathsf{Inst}_\Sigma(r)|.$$

Proof. Because $0 \leq \mathsf{ver}_{\Sigma,r}(\omega), \mathsf{fal}_{\Sigma,r}(\omega) \leq \mathsf{Inst}_\Sigma(r)$, we have

$$\begin{aligned} f_{\Sigma,r}(\omega) &= \mathsf{ver}_{\Sigma,r}(\omega) - p \cdot \mathsf{app}_{\Sigma,r}(\omega) \\ &= \mathsf{ver}_{\Sigma,r}(\omega) - p \cdot (\mathsf{ver}_{\Sigma,r}(\omega) + \mathsf{fal}_{\Sigma,r}(\omega)) \\ &= (1-p) \cdot \mathsf{ver}_{\Sigma,r}(\omega) - \underbrace{p \cdot \mathsf{fal}_{\Sigma,r}(\omega)}_{\geq 0} \\ &\leq (1-p) \cdot \mathsf{ver}_{\Sigma,r}(\omega) \\ &\leq (1-p) \cdot |\mathsf{Inst}_\Sigma(r)| \end{aligned}$$

as well as

$$\begin{aligned}
f_{\Sigma,r}(\omega) &= \mathsf{ver}_{\Sigma,r}(\omega) - p \cdot \mathsf{app}_{\Sigma,r}(\omega) \\
&= \mathsf{ver}_{\Sigma,r}(\omega) - p \cdot (\mathsf{ver}_{\Sigma,r}(\omega) + \mathsf{fal}_{\Sigma,r}(\omega)) \\
&= -p \cdot \mathsf{fal}_{\Sigma,r}(\omega) + \underbrace{(1-p) \cdot \mathsf{ver}_{\Sigma,r}(\omega)}_{\geq 0} \\
&\geq -p \cdot \mathsf{fal}_{\Sigma,r}(\omega) \\
&\geq -p \cdot |\mathsf{Inst}_{\Sigma}(r)|,
\end{aligned}$$

which proves the proposition. □

As a consequence of Proposition 3.4.9, for all $\omega \in \Omega(\Sigma)$, the quotient $\frac{f_{\Sigma,r}(\omega)}{|\mathsf{Inst}_{\Sigma}(r)|}$ lies in the interval $[-p, 1-p]$ and, with $p \in [0,1]$, it follows that

$$-1 \leq -p \leq \frac{f_r(\omega)}{|\mathsf{Inst}_{\Sigma}(r)|} \leq 1 - p \leq 1.$$

Thus, the fraction $\frac{f_{\Sigma,r}(\omega)}{|\mathsf{Inst}_{\Sigma}(r)|}$ is normalized to $[-1,1]$. This motivates the following notion of *standardized feature functions.*

Definition 3.4.10: Standardized Feature Function

Let Σ be a finite signature, let r be a probabilistic conditional in $\mathcal{RPCL}(\Sigma)$, and let $f_{\Sigma,r}\colon \Omega(\Sigma) \to \mathbb{R}$ be the feature function with respect to r according to Definition 3.4.3. Then, we call the mapping $\hat{f}_{\Sigma,r}\colon \Omega(\Sigma) \to [-1,1]$ with

$$\hat{f}_{\Sigma,r}(\omega) = \frac{f_{\Sigma,r}(\omega)}{|\mathsf{Inst}_{\Sigma}(r)|}$$

the *standardized feature function* with respect to r.

Note that $|\mathsf{Inst}_{\Sigma}(r)| \geq 1$ holds for all conditionals $r \in \mathcal{RPCL}(\Sigma)$ such that standardized feature functions are well-defined. Obviously, we have $\hat{f}_{\Sigma,r}(\omega) = -1$ only if $p = 1$ and $\hat{f}_{\Sigma,r}(\omega) = 1$ only if $p = 0$. That is, for non-factual conditionals $r = \delta[p]$ with $p \in (0,1)$, the stricter condition $\hat{f}_{\Sigma,r}(\omega) \in (-1,1)$ holds.

The standardized feature function $\hat{f}_{\Sigma,r}(\omega)$, which we consider here, differs from the *normalized feature function* introduced in [Finthammer, 2012]. The *normalized feature function* is defined with respect to a set of conditionals $\mathcal{B} = \{r_1, \ldots, r_n\}$ by

$$\tilde{f}_{\Sigma,r_i}(\omega) = \frac{f_{\Sigma,r_i}(\omega) + p_i \cdot |\mathsf{Inst}_{\Sigma}(r_i)|}{\sum_{i=1}^{n} |\mathsf{Inst}_{\Sigma}(r_i)|} \in [0,1], \quad \text{for} \qquad i = 1, \ldots, n, \tag{3.14}$$

where p_i is the probability of the conditional r_i.

The most obvious difference between the standardized feature function $\hat{f}_{\Sigma,r}$ and the normalized feature function $\tilde{f}_{\Sigma,r}$ is that $\hat{f}_{\Sigma,r}$ is normalized to $[-1,1]$ while $\tilde{f}_{\Sigma,r}$

ω	$f_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$	$\hat{f}_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$	$\tilde{f}_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$
$B(a)B(b)B(c)F(a)F(b)F(c)$	0.3	0.100	1
$B(a)B(b)B(c)F(a)F(b)\overline{F(c)}$	-0.7	-0.233	0.667
$B(a)B(b)B(c)F(a)\overline{F(b)}\,\overline{F(c)}$	-1.7	-0.567	0.333
$B(a)B(b)B(c)\overline{F(a)}\,\overline{F(b)}\,\overline{F(c)}$	-2.7	-0.900	0
$B(a)B(b)\overline{B(c)}F(a)F(b)F(c)$	0.2	0.067	0.967
$B(a)B(b)\overline{B(c)}F(a)\overline{F(b)}F(c)$	-0.8	-0.267	0.633
$B(a)B(b)\overline{B(c)}\,\overline{F(a)}\,\overline{F(b)}F(c)$	-1.8	-0.600	0.300
$B(a)\overline{B(b)}\,\overline{B(c)}F(a)F(b)F(c)$	0.1	0.033	0.933
$B(a)\overline{B(b)}\,\overline{B(c)}\,\overline{F(a)}F(b)F(c)$	-0.9	-0.300	0.600
$\overline{B(a)}\,\overline{B(b)}\,\overline{B(c)}F(a)F(b)F(c)$	0	0	0.900

Table 3.4: Feature function values $f_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$, standardized feature function values $\hat{f}_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$, and normalized feature function values $\tilde{f}_{\Sigma_{\mathsf{bf}},r_{\mathsf{bf}}}(\omega)$ of the possible worlds $\omega \in \Omega(\Sigma_{\mathsf{bf}})$ with respect to the conditional $r_{\mathsf{bf}} = (\mathsf{Flies}(X)|\mathsf{Bird}(X))[0.9]$ and $|\mathsf{Const}_{\Sigma_{\mathsf{bf}}}| = 3$. One representative of each equivalence class $[\omega]_{\sim_{\Sigma_{\mathsf{bf}},\{r_{\mathsf{bf}}\}}} \in \Omega_{\sim_{\Sigma_{\mathsf{bf}},\{r_{\mathsf{bf}}\}}}(\Sigma)$ is shown. Standardized and normalized feature function values are rounded to three digits.

is normalized to $[0,1]$. Another important difference is that the normalized feature function $\tilde{f}_{\Sigma,r}$ is defined with respect to a set of conditionals $\mathcal{B}$ so that $\tilde{f}_{\Sigma,r}$ depends on the other conditionals $r' \in \mathcal{B}$ with $r' \neq r$ as well. This is because the numbers of instances of the conditionals in $\mathcal{B}$ influence the denominator of the normalized feature function. Standardized feature functions can be calculated for each conditional independently instead.

Example 3.4.11

We consider the signature $\Sigma_{\mathsf{bf}} = (\mathsf{Const}_{\mathsf{bf}}, \mathsf{Pred}_{\mathsf{bf}})$ given by (cf. Example 3.4.6)

$$\mathsf{Const}_{\mathsf{bf}} = \{a, b, c\} \quad \text{and} \quad \mathsf{Pred}_{\mathsf{bf}} = \{\mathsf{Bird}/1,\ \mathsf{Flies}/1\}.$$

The standardized feature function values of the possible worlds in $\Omega(\Sigma_{\mathsf{bf}})$ with respect to the conditional $r_{\mathsf{bf}} = (\mathsf{Flies}(X)|\mathsf{Bird}(X))[0.9]$ are shown in Table 3.4. For comparison, the feature function values according to Definition 3.4.3 are recalled from Table 3.3 and also the normalized feature function values are given.

In the remainder of this thesis we will rely on the standardized feature functions but most results will hold for feature functions as defined in Definition 3.4.3 as well.

▶ Alternative Semantics of Open Probabilistic Conditionals

The aggregating semantics is not the only possible way of giving open probabilistic conditionals a formal meaning. Here, we briefly discuss two other semantics and argue why we have decided to consider the aggregating semantics in this thesis. The first alternative to the aggregating semantics is the naïve grounding of open probabilistic conditionals before interpreting them.

Definition 3.4.12: Grounding Semantics

(cf. [Kern-Isberner and Thimm, 2010])

Let Σ be a finite signature, and let $r \in \mathcal{RPCL}(\Sigma)$ be a (possibly open) probabilistic conditional. A probability distribution $\mathcal{P}\colon 2^{\Omega(\Sigma)} \to [0,1]$ is a *probabilistic model* of r *under the grounding semantics*, written

$$\mathcal{P} \models_{\mathsf{gr}} r, \quad \text{iff} \quad \mathcal{P} \models r' \quad \text{for all} \quad r' \in \mathsf{Inst}_{\Sigma}(r).$$

The grounding semantics interprets an open conditional r as a schema for all of its instances in $\mathsf{Inst}_{\Sigma}(r)$. In plain words, r is an abbreviation for the set of closed conditionals $\mathsf{Inst}_{\Sigma}(r)$. Actually, understanding expressions with free variables as schemata is in wide use in *knowledge representation and reasoning* (cf. e.g., *answer set programming* [Gelfond and Lifschitz, 1988; Gelfond, 2008] and *Reiter's default logic* [Reiter, 1980]) but it heavily depends on the application whether this view on open expressions makes sense. The authors of [Kern-Isberner and Thimm, 2010] give the following example which speaks against the grounding semantics.

Example 3.4.13: *(cf. [Kern-Isberner and Thimm, 2010])*

Let Σ_{el} be a signature with $\{\mathsf{clyde}, \mathsf{fred}\} \subseteq \mathsf{Const}_{\Sigma_{\mathsf{el}}}$ and

$$\mathsf{Pred}_{\Sigma_{\mathsf{el}}} = \{\mathsf{Likes}/2, \mathsf{Elephant}/1, \mathsf{Keeper}/1\}.$$

The probabilistic conditionals

$$\begin{aligned} r_1 &= (\mathsf{Likes}(X,Y) \mid \mathsf{Elephant}(X) \wedge \mathsf{Keeper}(Y))[0.6], \\ r_2 &= (\mathsf{Likes}(X,\mathsf{fred}) \mid \mathsf{Elephant}(X) \wedge \mathsf{Keeper}(\mathsf{fred}))[0.4], \\ r_3 &= (\mathsf{Likes}(\mathsf{clyde},\mathsf{fred}) \mid \mathsf{Elephant}(\mathsf{clyde}) \wedge \mathsf{Keeper}(\mathsf{fred}))[0.7], \end{aligned}$$

state that, in general, elephants like their zoo keepers with probability 0.6 (r_1), while the particular zoo keeper Fred is liked less by the elephants (r_2) despite of elephant Clyde that has an exceptional partiality for Fred (r_3). While, on an informal level, it is reasonable to accept all these conditionals at the same time, under the grounding semantics any set of conditionals including r_1, r_2, and r_3 is inconsistent. For instance, a probability distribution $\mathcal{P}$ which models r_1 under the grounding semantics models its instance $r'_1 \in \mathsf{Inst}_{\Sigma_{\mathsf{el}}}(r)$ given by

$$r'_1 = (\mathsf{Likes}(\mathsf{clyde}, \mathsf{fred}) \mid \mathsf{Elephant}(\mathsf{clyde}) \wedge \mathsf{Keeper}(\mathsf{fred}))[0.6]$$

in terms of the conditional probability

$$\mathcal{P}(\mathsf{Likes}(\mathsf{clyde}, \mathsf{fred}) \mid \mathsf{Elephant}(\mathsf{clyde}) \wedge \mathsf{Keeper}(\mathsf{fred})) = 0.6.$$

Then, $\mathcal{P}$ cannot model r_3 as well, as it differs from r'_1 in its probability value only (0.7 in contrast to 0.6). Hence, a probability distribution $\mathcal{P}$ cannot model both r'_1 and r_3 under the grounding semantics. Note that under the aggregating semantics instances of r_1 which are obtained by instantiating the tuple (X, Y) with other constants than $(\mathsf{clyde}, \mathsf{fred})$ can balance out the fact that Clyde likes Fred with probability 0.7 so that r_1 and r_3 not necessarily contradict each other.

The problem of inconsistencies that occur during the common grounding process of several conditionals as observed in Example 3.4.13 is discussed and partly solved in [Loh et al., 2010]. However, it remains a problem that the intended meaning of open conditionals is often misrepresented by the grounding semantics. For instance, in Example 3.4.13, the second conditional r_2 shall not state that *every* elephant likes the zoo keeper Fred less than an average zoo keeper, as it is realized in the grounding semantics, but that this differentiation holds in some kind of average. The idea of averaging probabilities cannot be captured by the grounding semantics, though, which is in contrast to the aggregating semantics (cf. the paragraph on *statistical and subject aspects of the aggregating semantics*).

The aggregating semantics not only introduces the concept of averaging probabilities but also amplifies the grounding semantics in the following sense: If a probability distribution $\mathcal{P}$ is a model of a conditional r under the grounding semantics, then $\mathcal{P}$ is also a model of r under the aggregating semantics but the reverse direction does not necessarily hold.

Proposition 3.4.14: Aggregating Versus Grounding Semantics
Let Σ be a finite signature, let r be a probabilistic conditional from $\mathcal{RPCL}(\Sigma)$, and let $\mathcal{P} : 2^{\Omega(\Sigma)} \to [0, 1]$ be a probability distribution. Then, $\mathcal{P} \models_{\mathsf{gr}} r$ implies $\mathcal{P} \models r$ but $\mathcal{P} \models r$ does not imply $\mathcal{P} \models_{\mathsf{gr}} r$.

Proof. Let $r = (B|A)_{\mathsf{CS}}[p]$. If $\mathcal{P} \models_{\mathsf{gr}} r$, then, by definition, $\mathcal{P}$ is a probabilistic Σ-model of $\mathsf{Inst}_\Sigma(r)$, i.e., $\mathcal{P} \models \mathsf{Inst}_\Sigma(r)$. Thus,

$$\sum_{\omega \in \Omega(\Sigma)} f_{\Sigma, r'}(\omega) \cdot \mathcal{P}(\omega) = 0$$

holds for all $r' \in \mathsf{Inst}_\Sigma(r)$. Further, we have

$$\begin{aligned} f_{\Sigma,r}(\omega) &= \mathsf{ver}_{\Sigma,r}(\omega) - p \cdot \mathsf{app}_{\Sigma,r}(\omega) \\ &= \sum_{r' \in \mathsf{Inst}_\Sigma(r)} \mathsf{ver}_{\Sigma,r'}(\omega) - p \cdot \sum_{r' \in \mathsf{Inst}_\Sigma(r)} \mathsf{app}_{\Sigma,r'}(\omega) \\ &= \sum_{r' \in \mathsf{Inst}_\Sigma(r)} (\mathsf{ver}_{\Sigma,r'}(\omega) - p \cdot \mathsf{app}_{\Sigma,r'}(\omega)) = \sum_{r' \in \mathsf{Inst}_\Sigma(r)} f_{\Sigma,r'}(\omega) \end{aligned}$$

for all $\omega \in \Omega(\Sigma)$. Consequently,

$$\begin{aligned} \sum_{\omega \in \Omega(\Sigma)} f_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) &= \sum_{\omega \in \Omega(\Sigma)} \left(\sum_{r' \in \mathsf{Inst}_\Sigma(r)} f_{\Sigma,r'}(\omega) \right) \cdot \mathcal{P}(\omega) \\ &= \sum_{r' \in \mathsf{Inst}_\Sigma(r)} \sum_{\omega \in \Omega(\Sigma)} f_{\Sigma,r'}(\omega) \cdot \mathcal{P}(\omega) = \sum_{\omega \in \Omega(\Sigma)} 0 = 0 \end{aligned}$$

follows. In addition, from $\mathcal{P}(A') > 0$ for $(B'|A')[p] \in \mathsf{Inst}_\Sigma(r)$ it follows that

$$\sum_{(B'|A')[p] \in \mathsf{Inst}_\Sigma(r)} \mathcal{P}(A') > 0$$

holds, too. Altogether, we have $\mathcal{P} \models r$.

For the proof that $\mathcal{P} \models r$ does not imply $\mathcal{P} \models \mathsf{Inst}_\Sigma(r)$, Example 3.4.13 serves as a counterexample. A simpler counterexample is the following. Let $\Sigma_{\mathsf{ex}} = (\{a, b\}, \{B/1\})$ be a signature, and let $r_{\mathsf{ex}} = (B(X)|\top)[0.8] \in \mathcal{RPCL}(\Sigma_{\mathsf{ex}})$. Then, the probability distribution $\mathcal{P}_{\mathsf{ex}}$ defined over $\Omega(\Sigma_{\mathsf{ex}})$ which is given by

$$\begin{aligned} &\mathcal{P}_{\mathsf{ex}}(B(a)B(b)) = 0.7, && \mathcal{P}_{\mathsf{ex}}(B(a)\overline{B(b)}) = 0.2, \\ &\mathcal{P}_{\mathsf{ex}}(\overline{B(a)}B(b)) = 0, && \mathcal{P}_{\mathsf{ex}}(\overline{B(a)}\,\overline{B(b)}) = 0.1, \end{aligned}$$

is a Σ_{ex}-model of r_{ex} with respect to the aggregating semantics because

$$\begin{aligned} &\frac{\mathcal{P}_{\mathsf{ex}}(B(a)) + \mathcal{P}_{\mathsf{ex}}(B(b))}{2 \cdot \mathcal{P}_{\mathsf{ex}}(\top)} \\ &= \frac{1}{2} \cdot (\mathcal{P}_{\mathsf{ex}}(B(a)) + \mathcal{P}_{\mathsf{ex}}(B(b))) \\ &= \frac{1}{2} \cdot \Big(\mathcal{P}_{\mathsf{ex}}(B(a)\overline{B(b)}) + 2 \cdot \mathcal{P}_{\mathsf{ex}}(B(a)B(b)) + \mathcal{P}_{\mathsf{ex}}(\overline{B(a)}B(b)) \Big) \\ &= \frac{1}{2} \cdot (0.2 + 2 \cdot 0.7 + 0) = 0.8. \end{aligned}$$

On the other side, $\mathcal{P}_{\mathsf{ex}} \not\models (B(a)|\top)[0.8]$ because

$$\mathcal{P}_{\mathsf{ex}}(B(a)|\top) = \mathcal{P}_{\mathsf{ex}}(B(a)B(b)) + \mathcal{P}_{\mathsf{ex}}(B(a)\overline{B(b)}) = 0.7 + 0.2 = 0.9 \neq 0.8.$$

As a consequence, because $(B(a)|\top)[0.8] \in \mathsf{Inst}_{\Sigma_{\mathsf{ex}}}(r_{\mathsf{ex}})$, the probability distribution $\mathcal{P}_{\mathsf{ex}}$ is not a Σ-model of $\mathsf{Inst}_{\Sigma_{\mathsf{ex}}}(r_{\mathsf{ex}})$. Thus, $\mathcal{P}_{\mathsf{ex}} \not\models_{\mathsf{gr}} r_{\mathsf{ex}}$ follows. ☐

The second alternative to the aggregating semantics which we want to draw attention to is the *averaging semantics* [Kern-Isberner and Thimm, 2010]. The averaging semantics demands the arithmetic mean of the conditional probabilities of the instances of a conditional to equal the conditional's probability.

Definition 3.4.15: Averaging Semantics

(cf. [Kern-Isberner and Thimm, 2010; Thimm, 2012])

Let Σ be a finite signature, and let $r \in \mathcal{RPCL}(\Sigma)$ with $r = (B|A)_{\mathsf{CS}}[p]$ be a (possibly open) probabilistic conditional. A probability distribution $\mathcal{P}\colon 2^{\Omega(\Sigma)} \to [0,1]$ is a *probabilistic model* of r *under the averaging semantics*, written

$$\mathcal{P} \models_{\mathsf{av}} r, \quad \text{iff} \quad \frac{1}{|\mathsf{Inst}_\Sigma(r)|} \cdot \sum_{(B'|A')[p] \in \mathsf{Inst}_\Sigma(r)} \mathcal{P}(B'|A') = p. \tag{3.15}$$

Like the aggregating and the grounding semantics, the averaging semantics properly generalizes the interpretation of closed probabilistic conditionals via conditional probabilities. Also its interpretation of open conditionals via the arithmetic mean of conditional probabilities is intuitive, especially in view of our remarks on how the conditionals in Example 3.4.13 have to be understood. A drawback of the averaging semantics is that the condition (3.15) is nonlinear, though. This is a problem when focusing on probabilistic models which maximize the entropy like in Chapter 4. While calculating a probabilistic model with maximal entropy under the aggregating semantics is a convex optimization problem and, therefore, has a unique solution, the respective problem with respect to the averaging semantics is non-convex, and it is not guaranteed that there is a unique probabilistic model which maximizes entropy. Hence, in the context of maximum entropy reasoning, the aggregating semantics turns out to be more convenient than the averaging semantics.

Note that it is conceivable to construe further semantics of open probabilistic conditionals from other means like the geometric or the harmonic mean as well. However, at least for the geometric and the harmonic mean this would also defeat the uniqueness of the probabilistic model at maximum entropy.

4 Relational Maximum Entropy Reasoning

The *principle of maximum entropy* (MaxEnt) [Jaynes, 1957a,b; Shannon, 1949; Paris, 1998] constitutes a valuable methodology for probabilistic commonsense reasoning by adding missing information to probabilistic conditional knowledge bases in an information theoretically optimal way [Gärdenfors, 1988]. This leads, in some sense, to most cautious reasoning. The main goal of this chapter is to develop the theoretical basis for drawing inferences from a probabilistic knowledge base at maximum entropy. In Section 4.1, we build the formal basis of this chapter by introducing *knowledge bases* and the *probabilistic inference task*. Then, in Section 4.2, we define the *maximum entropy model* and specify *maximum entropy inferences*. In Section 4.3, we discuss the *dual maximum entropy problem* which allows us to formulate a product representation of MaxEnt probabilities. This product representation will be essential for lifted inference at maximum entropy as it allows us to circumvent the computation of MaxEnt probabilities of single possible worlds when drawing inferences. Finally, we make an excursion and relate the principle of maximum entropy to *Markov Logic Networks* in Section 4.4.

4.1 Probabilistic Knowledge Representation and Reasoning

The main subject of this thesis is *drawing probabilistic inferences* from relational (resp. description-logical) probabilistic knowledge bases. Drawing inferences is one of the major tasks in *uncertain reasoning* (cf. e.g., [Griffiths and Yuille, 2012; Pearl, 2009; MacKay, 2003]). The subdiscipline which is concerned with probabilistic uncertain reasoning in complex relational settings—with which we also deal in this thesis—is known as *statistical relational AI (StarAI)* [Getoor and Taskar, 2007; De Raedt et al., 2016]. In contrast to common approaches in StarAI where one usually assumes that a probability distribution is given and drawing inferences means calculating the probabilities of specific events (cf. [De Raedt et al., 2016], Chapter 6), here we begin with a usually "incomplete" *knowledge base* which does not specify a distinct probabilistic model. The *probabilistic (inductive) inference task* consists of two steps then: (1) Calculating a probabilistic model of the knowledge base, and (2) drawing inferences based on this model.

In this section, we formally define probabilistic knowledge bases and their models. We show that the existence of probabilistic models may depend on the domain size, and we define the *probabilistic inductive inference task* that is considered in this thesis.

▶ Probabilistic Knowledge Representation

In this thesis, we aim at probabilistic *knowledge-based* reasoning. Hence, the notion of *knowledge bases* is central for us. Here, a *knowledge base* $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ is a tuple that consists of a finite set of sentences $\mathcal{F}_{\mathcal{R}} \subseteq \mathcal{RL}^S(\Sigma)$ representing factual knowledge, and a finite set of non-factual probabilistic conditionals $\mathcal{B}_{\mathcal{R}} \subseteq \mathcal{RPCL}(\Sigma)$ which formalize defeasible beliefs. Recall that a conditional $r \in \mathcal{RPCL}(\Sigma)$ is non-factual if its probability is neither 0 nor 1. Therewith, we have a clear separation between what the reasoner believes is certain ($\mathcal{F}_{\mathcal{R}}$) versus uncertain ($\mathcal{B}_{\mathcal{R}}$).

Definition 4.1.1: Knowledge Base *(cf. e.g., Wilhelm et al. [2017b])*

Let Σ be a finite signature. A *knowledge base* is a tuple $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ consisting of a finite set of sentences $\mathcal{F}_{\mathcal{R}} \subseteq \mathcal{RL}^S(\Sigma)$ and a finite set of non-factual probabilistic conditionals $\mathcal{B}_{\mathcal{R}} \subseteq \mathcal{RPCL}(\Sigma)$. The set of all knowledge bases over Σ is denoted with $\mathfrak{R}(\Sigma)$. Further, a knowledge base $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ is a *sub-knowledge base* of a knowledge base $\mathcal{R}' = (\mathcal{F}'_{\mathcal{R}'}, \mathcal{B}'_{\mathcal{R}'})$, denoted by $\mathcal{R} \sqsubseteq \mathcal{R}'$, if both $\mathcal{F}_{\mathcal{R}} \subseteq \mathcal{F}'_{\mathcal{R}'}$ and $\mathcal{B}_{\mathcal{R}} \subseteq \mathcal{B}'_{\mathcal{R}'}$ holds.

We typically refer to the set of uncertain beliefs $\mathcal{B}_{\mathcal{R}}$ of a knowledge base $\mathcal{R}$ by enumerating the conditionals in $\mathcal{B}_{\mathcal{R}}$. That is, we write $\mathcal{B}_{\mathcal{R}} = \{r_1, \dots, r_n\}$ with $r_i = (\psi_i|\phi_i)_{\mathsf{cs}_i}[p_i]$ for $i = 1, \dots, n$. In particular, we allocate $n = |\mathcal{B}_{\mathcal{R}}|$ for the number of conditionals in $\mathcal{B}_{\mathcal{R}}$ for the rest of this thesis.

The decision to exclude factual conditionals from the set $\mathcal{B}_{\mathcal{R}}$ is justified by the fact that factual conditionals are semantically equivalent to specific sentences in $\mathcal{RL}^S(\Sigma)$ up to the condition

$$\sum_{\omega \in \Omega(\Sigma)} \mathsf{app}_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) > 0 \tag{4.1}$$

which is the claim that the denominator of the fraction in the definition of the aggregating semantics must be positive such that the fraction is well-defined (cf. (3.11)).

More precisely, factual conditionals can be mimicked by sentences in $\mathcal{F}_{\mathcal{R}}$ according to the following proposition.

Proposition 4.1.2: Factual Conditionals Versus Sentences
Let Σ be a finite signature, let $r \in \mathcal{RPCL}(\Sigma)$ with $r = (\psi|\phi)_{\mathsf{CS}}[p]$ be a factual probabilistic conditional, i.e., $p \in \{0,1\}$, and let $\mathcal{P}\colon \Omega(\Sigma) \to [0,1]$ be a probability distribution. We define the classical counterpart of the conditional r in terms of material implication $\rho'(r) \in \mathcal{RL}(\Sigma)$ by

$$\rho'(r) = \begin{cases} \phi \Rightarrow \psi & \text{if} \quad p = 1 \\ \phi \Rightarrow \neg\psi & \text{if} \quad p = 0 \end{cases},$$

and its universally quantified closure by

$$\rho(r) = \forall_{\mathsf{CS}} \mathsf{FreeVar}(\rho'(r)).\ \rho'(r) \in \mathcal{RL}^S(\Sigma).$$

That is, $\rho(r)$ is $\rho'(r)$ where all free variables in $\rho'(r)$ are bounded by universal quantification. Then,

$$\mathcal{P} \models r \quad \text{iff} \quad \mathcal{P} \models \rho(r) \quad \text{and} \quad \sum_{\omega \in \Omega(\Sigma)} \mathsf{app}_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) > 0.$$

Proof. The feature function values of factual conditionals $r = (\psi|\phi)_{\mathsf{CS}}[p]$ with $p \in \{0,1\}$ reduce to $f_{\Sigma,r}(\omega) = \mathsf{ver}_{\Sigma,r}(\omega)$ in case of $p = 0$ and $f_{\Sigma,r}(\omega) = -\mathsf{fal}_{\Sigma,r}(\omega)$ in case of $p = 1$. Let $p = 0$. Then, for all $\omega \in \Omega(\Sigma)$ we have

$$f_{\Sigma,r}(\omega) = \mathsf{ver}_{\Sigma,r}(\omega) - 0 \cdot \mathsf{app}_{\Sigma,r}(\omega) = \mathsf{ver}_{\Sigma,r}(\omega),$$

and, according to (3.11) and provided that (4.1) holds,

$$\begin{aligned}
\mathcal{P} \models_{\Sigma} r \quad & \text{iff} \quad \sum_{\omega \in \Omega(\Sigma)} \mathsf{ver}_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) = 0 \\
& \text{iff} \quad \forall \omega \in \Omega(\Sigma) :\ (\mathsf{ver}_{\Sigma,r}(\omega) = 0 \vee \mathcal{P}(\omega) = 0) \\
& \text{iff} \quad \forall \omega \in \Omega(\Sigma) :\ (\mathsf{ver}_{\Sigma,r}(\omega) > 0 \Rightarrow \mathcal{P}(\omega) = 0) \\
& \text{iff} \quad \forall \omega \in \Omega(\Sigma) :\ ((\exists (\psi'|\phi')[p] \in \mathsf{Inst}_{\Sigma}(r) : \omega \models_{\Sigma} \phi'\psi') \Rightarrow \mathcal{P}(\omega) = 0) \\
& \text{iff} \quad \forall (\psi'|\phi')[p] \in \mathsf{Inst}_{\Sigma}(r)\ \forall \omega \in \Omega(\Sigma) : (\omega \models_{\Sigma} \phi'\psi' \Rightarrow \mathcal{P}(\omega) = 0) \\
& \text{iff} \quad \forall (\psi'|\phi')[p] \in \mathsf{Inst}_{\Sigma}(r) : \mathcal{P}(\phi' \wedge \psi') = 0 \\
& \text{iff} \quad \mathcal{P}\big(\bigvee\nolimits_{(\psi'|\phi')[p] \in \mathsf{Inst}_{\Sigma}(r)} \phi'\psi'\big) = 0 \\
& \text{iff} \quad \mathcal{P}(\exists_{\mathsf{CS}} \mathsf{FreeVar}(r).\phi\psi) = 0 \\
& \text{iff} \quad \mathcal{P}(\neg\exists_{\mathsf{CS}} \mathsf{FreeVar}(r).\phi\psi) = 1
\end{aligned}$$

$$\begin{aligned}
&\text{iff} \quad \mathcal{P}(\forall_{\mathsf{CS}}\mathsf{FreeVar}(r).\phi \Rightarrow \neg\psi) = 1\\
&\text{iff} \quad \mathcal{P}(\forall_{\mathsf{CS}}\mathsf{FreeVar}(\phi \Rightarrow \neg\psi).\phi \Rightarrow \neg\psi) = 1\\
&\text{iff} \quad p = 0 \wedge \mathcal{P} \models_{\Sigma} \rho(r).
\end{aligned}$$

Hereby, note that $\mathsf{FreeVar}(r) = \mathsf{FreeVar}(\phi \Rightarrow \neg\psi)$. Let $p = 1$ now. Then,

$$f_{\Sigma,r}(\omega) = \mathsf{ver}_{\Sigma,r}(\omega) - 1 \cdot \mathsf{app}_{\Sigma,r}(\omega) = \mathsf{ver}_{\Sigma,r}(\omega) - (\mathsf{ver}_{\Sigma,r}(\omega) + \mathsf{fal}_{\Sigma,r}(\omega)) = -\mathsf{fal}_{\Sigma,r}(\omega)$$

and

$$\begin{aligned}
\mathcal{P} \models_{\Sigma} r \quad &\text{iff} \quad \sum_{\omega\in\Omega(\Sigma)} (-\mathsf{fal}_{\Sigma,r}(\omega)) \cdot \mathcal{P}(\omega) = 0\\
&\text{iff} \quad \sum_{\omega\in\Omega(\Sigma)} \mathsf{fal}_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) = 0\\
&\text{iff} \quad \forall\omega \in \Omega(\Sigma): \ (\mathsf{fal}_{\Sigma,r}(\omega) = 0 \vee \mathcal{P}(\omega) = 0)\\
&\text{iff} \quad \forall\omega \in \Omega(\Sigma): \ (\mathsf{fal}_{\Sigma,r}(\omega) > 0 \Rightarrow \mathcal{P}(\omega) = 0)\\
&\text{iff} \quad \forall\omega \in \Omega(\Sigma): \ ((\exists(\psi'|\phi')[p] \in \mathsf{Inst}_{\Sigma}(r) : \omega \models_{\Sigma} \phi'\overline{\psi'}) \Rightarrow \mathcal{P}(\omega) = 0)\\
&\text{iff} \quad \forall(\psi'|\phi')[p] \in \mathsf{Inst}_{\Sigma}(r)\ \forall\omega \in \Omega(\Sigma) : (\omega \models_{\Sigma} \phi'\overline{\psi'} \Rightarrow \mathcal{P}(\omega) = 0)\\
&\text{iff} \quad \forall(\psi'|\phi')[p] \in \mathsf{Inst}_{\Sigma}(r) : \mathcal{P}(\phi'\overline{\psi'}) = 0\\
&\text{iff} \quad \mathcal{P}(\bigvee\nolimits_{(\psi'|\phi')[p]\in\mathsf{Inst}_{\Sigma}(r)} \phi'\overline{\psi'}) = 0\\
&\text{iff} \quad \mathcal{P}(\exists_{\mathsf{CS}}\mathsf{FreeVar}(r).\phi\overline{\psi}) = 0\\
&\text{iff} \quad \mathcal{P}(\neg\exists_{\mathsf{CS}}\mathsf{FreeVar}(r).\phi\overline{\psi}) = 1\\
&\text{iff} \quad \mathcal{P}(\forall_{\mathsf{CS}}\mathsf{FreeVar}(r).\phi \Rightarrow \psi) = 1\\
&\text{iff} \quad \mathcal{P}(\forall_{\mathsf{CS}}\mathsf{FreeVar}(\phi \Rightarrow \psi).\phi \Rightarrow \psi) = 1\\
&\text{iff} \quad p = 1 \wedge \mathcal{P} \models_{\Sigma} \rho(r).
\end{aligned}$$

Again, note that $\mathsf{FreeVar}(r) = \mathsf{FreeVar}(\phi \Rightarrow \psi)$. □

We will discuss the role of the condition (4.1) in the context of maximum entropy reasoning again. In particular, in Proposition 4.2.11 we will show that this condition is trivially satisfied by the *maximum entropy model* provided that the knowledge base is Σ-*p-consistent*. Hence, condition (4.1) is vacuous then, and we do not lose expressivity if we consider factual knowledge in $\mathcal{F}_{\mathcal{R}}$ instead of allowing for non-factual conditionals in $\mathcal{B}_{\mathcal{R}}$. We give some examples of knowledge bases now.

Example 4.1.3
Based on Example 2.2.2, we define three knowledge bases which will serve as running examples in the remainder of this thesis. The knowledge bases consist in part of probabilistic variants of conditionals which we have seen in Chapter 3 already (cf. e.g., Examples 3.3.2 and 3.4.2).

a) For our first knowledge base $\mathcal{R}_{\mathsf{bow}}$, we consider the already known signature $\Sigma_{\mathsf{bow}} = (\mathsf{Const}_{\Sigma_{\mathsf{bow}}}, \mathsf{Pred}_{\Sigma_{\mathsf{bow}}})$ with $\mathsf{Const}_{\Sigma_{\mathsf{bow}}} = \emptyset$ and

$$\mathsf{Pred}_{\Sigma_{\mathsf{bow}}} = \{\mathsf{Sunshine}/0,\ \mathsf{Rain}/0,\ \mathsf{Rainbow}/0\}$$

and define $\mathcal{R}_{\mathsf{bow}} = (\mathcal{F}_{\mathcal{R}_{\mathsf{bow}}}, \mathcal{B}_{\mathcal{R}_{\mathsf{bow}}})$ with $\mathcal{F}_{\mathcal{R}_{\mathsf{bow}}} = \emptyset$ and $\mathcal{B}_{\mathcal{R}_{\mathsf{bow}}} = \{r_1, r_2\}$ where

$$\begin{aligned} r_1 &= (\mathsf{Rainbow}|\mathsf{Sunshine} \wedge \mathsf{Rain})[0.6], \\ r_2 &= (\neg\mathsf{Rainbow}|\neg\mathsf{Sunshine} \vee \neg\mathsf{Rain})[0.99]. \end{aligned}$$

The knowledge base $\mathcal{R}_{\mathsf{bow}}$ states that there is a chance to observe a rainbow when it is rainy weather and the sun shines ($p_1 = 0.6$), but it is very likely to observe no rainbow when this premise does not hold ($p_2 = 0.99$).

b) Second, we consider any signature $\Sigma_{\mathsf{bfp}} = (\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}, \mathsf{Pred}_{\Sigma_{\mathsf{bfp}}})$ with $\mathsf{tweety} \in \mathsf{Const}_{\Sigma_{\mathsf{bfp}}}$ and

$$\mathsf{Pred}_{\Sigma_{\mathsf{bfp}}} = \{\mathsf{Bird}/1,\ \mathsf{Flies}/1,\ \mathsf{Penguin}/1\}.$$

The knowledge base $\mathcal{R}_{\mathsf{bfp}} = (\mathcal{F}_{\mathcal{R}_{\mathsf{bfp}}}, \mathcal{B}_{\mathcal{R}_{\mathsf{bfp}}})$ consists of the set of facts $\mathcal{F}_{\mathcal{R}_{\mathsf{bfp}}} = \{F_1, F_2\}$ and the set of beliefs $\mathcal{B}_{\mathcal{R}_{\mathsf{bfp}}} = \{r_1, r_2\}$ with

$$\begin{aligned} F_1 &= \forall X.(\mathsf{Penguin}(X) \Rightarrow \mathsf{Bird}(X)), \\ F_2 &= \mathsf{Penguin}(\mathsf{tweety}), \\ \\ r_1 &= (\mathsf{Flies}(X)|\mathsf{Bird}(X))[0.9], \\ r_2 &= (\neg\mathsf{Flies}(X)|\mathsf{Penguin}(X))[0.99], \end{aligned}$$

and refers to Example 3.3.2. $\mathcal{R}_{\mathsf{bfp}}$ states that all penguins are birds, Tweety is a penguin, birds are usually able to fly, namely with probability $p_1 = 0.9$, and penguins are very likely not able to fly ($p_2 = 0.99$).

c) The third knowledge base which we want to investigate is defined for any signature $\Sigma_{\mathsf{smk}} = (\mathsf{Const}_{\Sigma_{\mathsf{smk}}}, \mathsf{Pred}_{\Sigma_{\mathsf{smk}}})$ with

$$\mathsf{Pred}_{\Sigma_{\mathsf{smk}}} = \{\mathsf{Smokes}/1,\ \mathsf{Friends}/2\}$$

and is $\mathcal{R}_{\mathsf{smk}} = (\mathcal{F}_{\mathcal{R}_{\mathsf{smk}}}, \mathcal{B}_{\mathcal{R}_{\mathsf{smk}}})$ with $\mathcal{F}_{\mathcal{R}_{\mathsf{smk}}} = \emptyset$ and $\mathcal{B}_{\mathcal{R}_{\mathsf{smk}}} = \{r_1, r_2\}$ where

$$r_1 = (\mathsf{Friends}(Y,X)|\mathsf{Friends}(X,Y))[0.9],$$
$$r_2 = (\mathsf{Smokes}(X)|\mathsf{Friends}(X,Y) \wedge \mathsf{Smokes}(Y))_{\{X \neq Y\}}[0.8].$$

The knowledge base $\mathcal{R}_{\mathsf{smk}}$ states that being friends is usually a symmetric relation ($p_1 = 0.9$) and that persons who have a smoking friend usually smoke, too ($p_2 = 0.8$). We will consider $\mathcal{R}_{\mathsf{smk}}$ for different sets of constants $\mathsf{Const}_{\Sigma_{\mathsf{smk}}}$.

The semantics of knowledge bases is given by probability distributions over possible worlds.

Definition 4.1.4: Probabilistic Model of a Knowledge Base

(cf. [Kern-Isberner and Thimm, 2010])

Let Σ be a finite signature, and let $\mathcal{R} \in \mathfrak{R}(\Sigma)$ be a knowledge base with $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$. A probability distribution $\mathcal{P}\colon 2^{\Omega(\Sigma)} \to [0,1]$ is a *probabilistic* Σ*-model* of $\mathcal{R}$ iff both

- for all $\omega \in \Omega(\Sigma)$, $\omega \not\models_{\Sigma} \mathcal{F}_{\mathcal{R}}$ implies $\mathcal{P}(\omega) = 0$,
- for all $r \in \mathcal{B}_{\mathcal{R}}$, $\mathcal{P} \models r$, where $\mathcal{P} \models r$ is decided based on the aggregating semantics (cf. Definition 3.4.1).

The set of all probabilistic Σ-models of $\mathcal{R}$ is denoted with $\mathsf{probMod}_{\Sigma}(\mathcal{R})$.

The first condition of Definition 4.1.4 states that possible worlds $\omega \in \Omega(\Sigma)$ which violate at least one fact from $\mathcal{F}_{\mathcal{R}}$ are mapped to the probability value $\mathcal{P}(\omega) = 0$ by all probabilistic models $\mathcal{P} \in \mathsf{probMod}_{\Sigma}(\mathcal{R})$, hence, are classified as *impossible*. In other words, possible worlds have a positive probability only if they are models of $\mathcal{F}_{\mathcal{R}}$. The second condition states that probabilistic models of $\mathcal{R}$ have to model all conditionals in $\mathcal{B}_{\mathcal{R}}$ according to the aggregating semantics. That is, a probability distribution has to satisfy (3.11) for all conditionals in $\mathcal{B}_{\mathcal{R}}$ in order to be a model of $\mathcal{R}$.

The next proposition justifies the term *fact(ual knowledge)* for sentences in $\mathcal{F}_{\mathcal{R}}$ and shows that facts are certain events in probabilistic models of $\mathcal{R}$. That is, if $\mathcal{P} \in \mathsf{probMod}_{\Sigma}(\mathcal{R})$, then $\mathcal{P}(F) = 1$ holds for all facts $F \in \mathcal{F}_{\mathcal{R}}$. In other words, $\mathcal{P} \models F$ for all $F \in \mathcal{F}_{\mathcal{R}}$.

Proposition 4.1.5: Probabilistic Entailment of Facts
Let Σ be a finite signature, let $\mathcal{R} \in \mathfrak{R}(\Sigma)$ be a knowledge base with $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$, and let $\mathcal{P}$ be a probabilistic Σ-model of $\mathcal{R}$. Then, $\mathcal{P} \models F$ holds for all $F \in \mathcal{F}_{\mathcal{R}}$.

Proof. Let $F \in \mathcal{F}_{\mathcal{R}}$ be a fact from $\mathcal{R}$. For all $\omega \in \Omega(\Sigma)$, we have $\omega \models_{\Sigma} F$ if and only if $\omega \not\models_{\Sigma} \overline{F}$. For $\mathcal{P} \in \mathsf{probMod}_{\Sigma}(\mathcal{R})$, it follows that

$$\mathcal{P}(\overline{F}) = \sum_{\omega \in \Omega(\Sigma)\colon \omega \models_{\Sigma} \overline{F}} \mathcal{P}(\omega) = \sum_{\omega \in \Omega(\Sigma)\colon \omega \models_{\Sigma} \overline{F}} 0 = 0$$

and, further, $\mathcal{P}(F) = 1 - \mathcal{P}(\overline{F}) = 1 - 0 = 1$. Therewith, $\mathcal{P} \models F$ follows by the definition of probabilistic models (Definition 4.1.4). □

As a consequence of Proposition 4.1.5, “impossible” possible worlds, i.e., possible worlds $\omega \in \Omega(\Sigma)$ with $\omega \not\models_{\Sigma} \mathcal{F}_{\mathcal{R}}$, can be ignored in probabilistic reasoning because they do not contribute to the probabilities of probabilistic sentences or conditionals. For this reason, we differentiate between the *set of impossible worlds*

$$\begin{aligned} \Omega^0_{\mathcal{F}_{\mathcal{R}}}(\Sigma) &= \{\omega \in \Omega(\Sigma) \mid \omega \not\models_{\Sigma} \mathcal{F}_{\mathcal{R}}\} \\ &= \{\omega \in \Omega(\Sigma) \mid \omega \models_{\Sigma} \neg \textstyle\bigwedge_{F \in \mathcal{F}_{\mathcal{R}}} F\} \\ &= \{\omega \in \Omega(\Sigma) \mid \omega \models_{\Sigma} \textstyle\bigvee_{F \in \mathcal{F}_{\mathcal{R}}} \neg F\} \end{aligned} \tag{4.2}$$

and the set of *potentially positive worlds*

$$\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma) = \{\omega \in \Omega(\Sigma) \mid \omega \models_{\Sigma} \mathcal{F}_{\mathcal{R}}\} \tag{4.3}$$

relative to a knowledge base $\mathcal{R}$. Obviously, it holds that $\Omega(\Sigma) = \Omega^0_{\mathcal{F}_{\mathcal{R}}}(\Sigma) \,\dot\cup\, \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ for every knowledge base $\mathcal{R}$ (more precisely, for every set of facts $\mathcal{F}_{\mathcal{R}}$) where $\dot\cup$ denotes the disjoint union. As a consequence, every probabilistic Σ-model $\mathcal{P}$ of a knowledge base $\mathcal{R} \in \mathfrak{R}(\Sigma)$ is of the form

$$\mathcal{P}(\omega) \begin{cases} \geq 0 & \text{if} \quad \omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma) \\ = 0 & \text{if} \quad \omega \in \Omega^0_{\mathcal{F}_{\mathcal{R}}}(\Sigma) \end{cases}. \tag{4.4}$$

This enables us to restrict ourself to probability distributions which are defined over $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ and to assign the probability $\mathcal{P}(\omega) = 0$ to the remaining possible worlds $\omega \in \Omega^0_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ afterwards when computing probabilistic models of $\mathcal{R}$. To sum up, the semantics of probabilistic models (Definition 4.1.4) does not force potentially positive worlds to have a positive probability in general but they are the only candidates.

Example 4.1.6

We consider the signature $\Sigma_{\mathsf{bfp}} = (\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}, \mathsf{Pred}_{\Sigma_{\mathsf{bfp}}})$ (cf. Example 2.2.2) given by

$$\mathsf{Const}_{\Sigma_{\mathsf{bfp}}} = \{\mathsf{sally}, \mathsf{tweety}\} \quad \text{and} \quad \mathsf{Pred}_{\Sigma_{\mathsf{bfp}}} = \{\mathsf{Bird}/1,\ \mathsf{Flies}/1,\ \mathsf{Penguin}/1\}$$

and abbreviate $s = \mathsf{sally}$ and $t = \mathsf{tweety}$ as well as $B = \mathsf{Bird}$, $F = \mathsf{Flies}$, and $P = \mathsf{Penguin}$. A probability distribution $\mathcal{P}$ over $\Omega(\Sigma_{\mathsf{bfp}})$ is a model of the knowledge base $\mathcal{R}_{\mathsf{bfp}} = (\{F_1, F_2\}, \{r_1, r_2\})$ with (cf. Example 4.1.3)

$$\begin{aligned} F_1 &= \forall X.(\mathsf{Penguin}(X) \Rightarrow \mathsf{Bird}(X)), \\ F_2 &= \mathsf{Penguin}(\mathsf{tweety}), \end{aligned}$$

$$\begin{aligned} r_1 &= (\mathsf{Flies}(X) | \mathsf{Bird}(X))[0.9], \\ r_2 &= (\neg\mathsf{Flies}(X) | \mathsf{Penguin}(X))[0.99], \end{aligned}$$

iff

$$\begin{aligned} F_1 : \quad & \mathcal{P}((P(s) \Rightarrow B(s)) \wedge (P(t) \Rightarrow B(t))) = 1, \\ F_2 : \quad & \mathcal{P}(P(t)) = 1, \\ r_1 : \quad & \frac{\mathcal{P}(B(s)F(s)) + \mathcal{P}(B(t)F(t))}{\mathcal{P}(B(s)) + \mathcal{P}(B(t))} = 0.9, \\ & \text{and} \quad \mathcal{P}(B(s)) + \mathcal{P}(B(t)) > 0, \\ r_2 : \quad & \frac{\mathcal{P}(P(s)\overline{F(s)}) + \mathcal{P}(P(t)\overline{F(t)})}{\mathcal{P}(P(s)) + \mathcal{P}(P(t))} = 0.99, \\ & \text{and} \quad \mathcal{P}(P(s)) + \mathcal{P}(P(t)) > 0. \end{aligned}$$

The set of the impossible worlds $\Omega^0_{\mathcal{F}_{\mathcal{R}_{\mathsf{bfp}}}}(\Sigma_{\mathsf{bfp}})$ is given by

$$\begin{aligned} \Omega^0_{\mathcal{F}_{\mathcal{R}_{\mathsf{bfp}}}}(\Sigma_{\mathsf{bfp}}) &= \{\omega \in \Omega(\Sigma_{\mathsf{bfp}}) \mid \omega \notin \mathsf{Mod}_{\Sigma_{\mathsf{bfp}}}(\{F_1, F_2\}) \\ &= \{\omega \in \Omega(\Sigma_{\mathsf{bfp}}) \mid \omega \in \mathsf{Mod}_{\Sigma_{\mathsf{bfp}}}(P(S)\overline{B(s)} \vee P(t)\overline{B(t)} \vee \overline{P(t)})\}. \end{aligned}$$

The set of the potentially positive worlds is

$$\begin{aligned} \Omega_{\mathcal{F}_{\Sigma_{\mathsf{bfp}}}}(\Sigma_{\mathsf{bfp}}) &= \Omega_{\Sigma_{\mathsf{bfp}}} \setminus \Omega^0_{\mathcal{F}_{\Sigma_{\mathsf{bfp}}}}(\Sigma_{\mathsf{bfp}}) \\ &= \{\omega \in \Omega(\Sigma_{\mathsf{bfp}}) \mid \omega \in \mathsf{Mod}_{\Sigma_{\mathsf{bfp}}}((P(s) \Rightarrow B(s)) \\ &\qquad \wedge (P(t) \Rightarrow B(t)) \wedge P(t))\} \end{aligned}$$

$$\begin{aligned}
= \{&P(s)B(s)F(s)P(t)B(t)F(t),\ P(s)B(s)F(s)P(t)B(t)\overline{F(t)},\\
&P(s)B(s)\overline{F(s)}P(t)B(t)F(t),\ P(s)B(s)\overline{F(s)}P(t)B(t)\overline{F(t)},\\
&\overline{P(s)}B(s)F(s)P(t)B(t)F(t),\ \overline{P(s)}B(s)F(s)P(t)B(t)\overline{F(t)},\\
&\overline{P(s)}B(s)\overline{F(s)}P(t)B(t)F(t),\ \overline{P(s)}B(s)\overline{F(s)}P(t)B(t)\overline{F(t)},\\
&\overline{P(s)}\,\overline{B(s)}F(s)P(t)B(t)F(t),\ \overline{P(s)}\,\overline{B(s)}F(s)P(t)B(t)\overline{F(t)},\\
&\overline{P(s)}\,\overline{B(s)}\,\overline{F(s)}P(t)B(t)F(t),\ \overline{P(s)}\,\overline{B(s)}\,\overline{F(s)}P(t)B(t)\overline{F(t)}\}.
\end{aligned}$$

Having the concept of syntax splitting in mind (cf. [Parikh, 1999; Kern-Isberner and Brewka, 2017; Wilhelm et al., 2017b] and Example 3.2.4), the set $\Omega_{\mathcal{F}_{\Sigma_{\mathsf{bfp}}}}(\Sigma_{\mathsf{bfp}})$ can also be written as

$$\begin{aligned}
\Omega_{\mathcal{F}_{\Sigma_{\mathsf{bfp}}}}(\Sigma_{\mathsf{bfp}}) = \{\omega_1 \wedge \omega_2 \mid \omega_1 \in \{&P(s)B(s)F(s),\ P(s)B(s)\overline{F(s)},\\
&\overline{P(s)}B(s)F(s),\ \overline{P(s)}B(s)\overline{F(s)},\\
&\overline{P(s)}\,\overline{B(s)}F(s),\ \overline{P(s)}\,\overline{B(s)}\,\overline{F(s)}\},\\
\omega_2 \in \{&P(t)B(t)F(t),\ P(t)B(t)\overline{F(t)}\} \quad \}.
\end{aligned}$$

Only possible worlds in $\Omega_{\mathcal{F}_{\Sigma_{\mathsf{bfp}}}}(\Sigma_{\mathsf{bfp}})$ are possibly assigned a positive probability by a probabilistic Σ_{bfp}-model $\mathcal{P}$ of $\mathcal{R}_{\mathsf{bfp}}$.

A common precondition in probabilistic knowledge-based reasoning is the *consistency* of the underlying knowledge base. A knowledge base $\mathcal{R} \in \mathfrak{R}(\Sigma)$ is *consistent* if it has at least one model. The consistency of knowledge bases depends on the size of Const_Σ. The next paragraph is dedicated to this issue.

▶ Consistency and its Dependency on the Domain Size

Probabilistic knowledge-based reasoning is usually based on probabilistic models of a knowledge base such that the existence of a model is a necessary precondition, in particular for the probabilistic inductive inference task. If a knowledge base has a model, then the knowledge base is called *consistent*. More precisely, we have the following definition of *consistency* which takes the underlying signature into account.

Definition 4.1.7: Consistency of a Knowledge Base

(cf. e.g., [Kern-Isberner and Thimm, 2010])

Let Σ be a finite signature, and let $\mathcal{R} \in \mathfrak{R}(\Sigma)$ be a knowledge base. Then, $\mathcal{R}$ is called *consistent* with respect to Σ, or Σ-*consistent* for short, iff $\mathcal{R}$ has at least one probabilistic Σ-model, i.e., iff $\mathsf{probMod}_\Sigma(\mathcal{R}) \neq \emptyset$. Otherwise, $\mathcal{R}$ is called Σ-*inconsistent*.

Whether a knowledge base has a probabilistic Σ-model or not also depends on the *domain size* $k = |\mathsf{Const}_\Sigma|$, i.e., the number of constants in the underlying signature Σ. Recall that the term *domain* usually refers to semantical entities and not to syntactical constants. However, because relational logics $\mathcal{RL}(\Sigma)$ rely on a Herbrand semantics here and are subject to the *unique name assumption* (cf. Section 2.2), constants are in a one-to-one-correspondence to their interpretations so that it is well justified to call Const_Σ the *domain*. We show the domain-dependency of the consistency of knowledge bases by means of an example.

Example 4.1.8

We consider the knowledge base $\mathcal{R}_{\mathsf{ex}} = (\{F_1, F_2\}, \{r_1\})$ with

$$\begin{aligned} F_1 &= \phi(c), \\ F_2 &= \overline{\psi(c)}, \\ r_1 &= (\psi(X)|\phi(X))[0.5]. \end{aligned}$$

For $k = 1$, i.e., if $\mathsf{Const}_{\Sigma_{\mathsf{ex}}} = \{c\}$, the knowledge base $\mathcal{R}_{\mathsf{ex}}$ is inconsistent. A probabilistic model $\mathcal{P}$ of $\mathcal{R}_{\mathsf{ex}}$ would have to satisfy

$$\frac{\mathcal{P}(\phi(c) \wedge \psi(c))}{\mathcal{P}(\phi(c))} = 0.5 \tag{4.5}$$

in order to model r_1. However, because of F_2 we have $\mathcal{P}(\psi(c)) = 0$ and, thus, also $\mathcal{P}(\phi(c) \wedge \psi(c)) = 0$ so that (4.5) cannot hold.

For $k = 2$, the knowledge base $\mathcal{R}_{\mathsf{ex}}$ is consistent instead. Let $\mathsf{Const}_{\Sigma_{\mathsf{ex}}} = \{c, d\}$. Then $\mathcal{P}$ models r_1 if

$$\begin{aligned} 0.5 &= \frac{\mathcal{P}(\phi(c)\psi(c)) + \mathcal{P}(\phi(d)\psi(d))}{\mathcal{P}(\phi(c)) + \mathcal{P}(\phi(d))} \\ &= \frac{\mathcal{P}(\phi(c)\overline{\psi(c)}\phi(d)\psi(d))}{X} \end{aligned}$$

with

$$\begin{aligned} X = {} & 2 \cdot \mathcal{P}(\phi(c)\overline{\psi(c)}\phi(d)\psi(d)) + \mathcal{P}(\phi(c)\overline{\psi(c)}\,\overline{\phi(d)}\psi(d)) \\ & + \mathcal{P}(\phi(c)\overline{\psi(c)}\,\overline{\phi(d)}\,\overline{\psi(d)}) + 2 \cdot \mathcal{P}(\phi(c)\overline{\psi(c)}\phi(d)\overline{\psi(d)}), \end{aligned}$$

where we already exploited $\mathcal{P}(\overline{\phi(c)}) = \mathcal{P}(\psi(c)) = 0$. This holds for every probability distribution $\mathcal{P}$ with

$$\begin{aligned} &\mathcal{P}(\phi(c)\overline{\psi(c)}\phi(d)\psi(d)) > 0, && \mathcal{P}(\phi(c)\overline{\psi(c)}\,\overline{\phi(d)}\psi(d)) = 0, \\ &\mathcal{P}(\phi(c)\overline{\psi(c)}\,\overline{\phi(d)}\,\overline{\psi(d)}) = 0, && \mathcal{P}(\phi(c)\overline{\psi(c)}\phi(d)\overline{\psi(d)}) = 0. \end{aligned}$$

In this case, the constant d plays the counterpart to c in the evaluation of r_1. While c negates the belief that "ϕ usually leads to ψ," d confirms it, which results in the probability $p_1 = 0.5$ of the conditional r_1.

Example 4.1.8 shows that the probabilistic constraints of a knowledge base can require a minimal number of constants in order to be satisfied. In Example 4.1.8, the dependency on the domain size origins from the probability value p_1 of r_1. If p_1 was 0 instead of 0.5, for instance, the knowledge base $\mathcal{R}_{\mathsf{ex}}$ would be consistent for $k = 1$ already. The knowledge base $\mathcal{R}_{\mathsf{ex}}$ is a simple example where a probability necessitates a minimal number of domain elements. Usually, such dependencies are more complex and several probability values have to be considered in combination.

Another reason why a knowledge base $\mathcal{R}$ could be inconsistent is that logical constraints of expression in $\mathcal{R}$, i.e., facts or logical parts of conditionals, are inconsistent. For instance, this happens if $\mathsf{Mod}_\Sigma(\mathcal{F}_\mathcal{R}) = \emptyset$ or there is a conditional $r \in \mathcal{B}_\mathcal{R}$ which is not applicable, i.e., $\mathsf{app}_{\Sigma,r}(\omega) = 0$ for all $\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$.

Example 4.1.9

- It is clear that there are sets of facts $\mathcal{F}_\mathcal{R}$ with $\mathsf{Mod}_\Sigma(\mathcal{F}_\mathcal{R}) = \emptyset$ and which, thus, cause inconsistency of $\mathcal{R} = (\mathcal{F}_\mathcal{R}, \mathcal{B}_\mathcal{R})$. More interestingly, whether $\mathsf{Mod}_\Sigma(\mathcal{F}_\mathcal{R}) = \emptyset$ holds or not can also depend on the domain size $k = |\mathsf{Const}_\Sigma|$. To see this, consider $\phi \in \mathcal{RL}^S(\Sigma)$ with

  $$\phi = \exists X.A(X) \wedge \exists Y.\overline{A(Y)}.$$

  The sentence ϕ is satisfiable for $k > 1$ and unsatisfiable for $k = 1$.

- We consider the knowledge base $\mathcal{R}_{\mathsf{smk}}$ from Example 4.1.3(c) which involves the conditional

  $$r_2 = (\mathsf{Smokes}(X)|\mathsf{Friends}(X,Y) \wedge \mathsf{Smokes}(Y))_{\{X \neq Y\}}[0.8].$$

  If $|\mathsf{Const}_{\Sigma_{\mathsf{smk}}}| = 1$, then $\mathsf{app}_{\Sigma,r_2}(\omega) = 0$ holds for all $\omega \in \Omega_{\mathcal{F}_{\mathcal{R}_{\mathsf{smk}}}}(\Sigma)$ because $\mathsf{Inst}(r_2) = \emptyset$ in this case due to the domain constraint $X \neq Y$ which calls for at least two constants in Const_Σ.

We can put on record that the inconsistency of a knowledge base $\mathcal{R}$ can be caused by both conflicting probabilities and by unsatisfiable logical expressions. Both types of inconsistency can depend on the domain size $k = |\mathsf{Const}_\Sigma|$. Necessary but not sufficient conditions for the Σ-consistency of a knowledge base $\mathcal{R} \in \mathfrak{R}(\Sigma)$ are the condition $\mathsf{Mod}_\Sigma(\mathcal{F}_\mathcal{R}) \neq \emptyset$ and, for all conditionals $r \in \mathcal{B}_\mathcal{R}$, the existence of a possible world $\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$ with $\mathsf{app}_r(\omega) > 0$.

▶ Probabilistic Inductive Inference Task

Drawing inductive inferences from a knowledge base is one of the most important reasoning tasks in the field of knowledge representation and reasoning (KR). Here, drawing inferences means that we assume that a consistent knowledge base $\mathcal{R}$ is given and we ask for the conditionals which hold in a specific probabilistic model of $\mathcal{R}$ ("model-based inductive inference"). Formally, such inductive inferences are defined via relations between knowledge bases $\mathcal{R}$ and probabilistic conditionals r with respect to a model of $\mathcal{R}$ as follows.

Definition 4.1.10: Probabilistic Inductive Inference Relation

(cf. e.g., [Kern-Isberner and Thimm, 2010])

Let Σ be a finite signature, let $\mathcal{R} \in \mathfrak{R}(\Sigma)$ be a Σ-consistent knowledge base, let $\mathcal{P}_{\mathcal{R}} \in \mathsf{probMod}_{\Sigma}(\mathcal{R})$ be a probabilistic Σ-model of $\mathcal{R}$, and let $r \in \mathcal{RPCL}(\Sigma)$ be a probabilistic conditional. Then, r can be *inferred* from $\mathcal{R}$ with respect to $\mathcal{P}_{\mathcal{R}}$, written

$$\mathcal{R} \models_{\mathcal{P}} r \quad \text{iff} \quad \mathcal{P}_{\mathcal{R}} \models r.$$

We call the relation $\models_{\mathcal{P}}$ *(probabilistic inductive) inference relation* and the conditional r *query (conditional).*

The inference relation $\models_{\mathcal{P}}$ from Definition 4.1.10 can be extended to sentences from $\mathcal{RL}^S(\Sigma)$ in the obvious manner: Let $\phi \in \mathcal{RL}^S(\Sigma)$ be a sentence. Then,

$$\mathcal{R} \models_{\mathcal{P}} \phi \quad \text{iff} \quad \mathcal{R} \models_{\mathcal{P}} (\phi|\top)[1].$$

Consequently, we do not have to distinguish between query sentences and query conditionals because every query sentence can be expressed as a query conditional.

We discuss some basic properties of the inference relation $\models_{\mathcal{P}}$ which hold for every model $\mathcal{P}$ of $\mathcal{R}$. To start with, all the facts and conditionals from $\mathcal{R}$ can be inferred from $\mathcal{R}$ via $\models_{\mathcal{P}}$ because $\models_{\mathcal{P}}$ is a model-based inference relation. This property is known as *direct inference.*

Proposition 4.1.11: Direct Inference

Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a Σ-consistent knowledge base, and let $\mathcal{P} \in \mathsf{probMod}_{\Sigma}(\mathcal{R})$ be a probabilistic Σ-model of $\mathcal{R}$. Then,

- $\mathcal{R} \models_{\mathcal{P}} \phi$ for all $\phi \in \mathcal{F}_{\mathcal{R}}$.
- $\mathcal{R} \models_{\mathcal{P}} r$ for all $r \in \mathcal{B}_{\mathcal{R}}$.

Proof. This directly follows from the definition of probabilistic models: We have $\mathcal{R} \models_{\mathcal{P}} \phi$ (resp. $\mathcal{R} \models_{\mathcal{P}} r$) if and only if $\mathcal{P} \models \phi$ (resp. $\mathcal{P} \models r$) which holds for all $\phi \in \mathcal{F}_{\mathcal{R}}$ (resp. $r \in \mathcal{B}_{\mathcal{R}}$) because $\mathcal{P}$ models $\mathcal{F}_{\mathcal{R}}$ and $\mathcal{B}_{\mathcal{R}}$. $\square$

Also sentences which can be classically deduced from $\mathcal{F}_{\mathcal{R}}$ are inferable from $\mathcal{R}$. This property is called *supraclassicality.*

Proposition 4.1.12: Supraclassicality

Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a Σ-consistent knowledge base, and let $\mathcal{P} \in \mathsf{probMod}_{\Sigma}(\mathcal{R})$ be a probabilistic Σ-model of $\mathcal{R}$. Further let $\phi \in \mathcal{RL}^S(\Sigma)$ be a sentence. Then,

$$\mathcal{F}_{\mathcal{R}} \models_{\Sigma} \phi \quad \text{implies} \quad \mathcal{R} \models_{\mathcal{P}} \phi.$$

Proof. From $\mathcal{F}_{\mathcal{R}} \models_{\Sigma} \phi$ it follows that $\mathsf{Mod}_{\Sigma}(\mathcal{F}_{\mathcal{R}}) \subseteq \mathsf{Mod}_{\Sigma}(\phi)$ holds. Since $\mathcal{P}(\omega) = 0$ for all $\omega \notin \mathsf{Mod}_{\Sigma}(\mathcal{F}_{\mathcal{R}})$, we have

$$\sum_{\omega \in \Omega(\Sigma)\colon \omega \in \mathsf{Mod}_{\Sigma}(\mathcal{F}_{\mathcal{R}})} \mathcal{P}(\omega) = 1$$

and, hence,

$$\mathcal{P}(\phi) = \sum_{\omega \in \Omega(\Sigma)\colon \omega \in \mathsf{Mod}_{\Sigma}(\phi)} \mathcal{P}(\omega) \geq \sum_{\omega \in \Omega(\Sigma)\colon \omega \in \mathsf{Mod}_{\Sigma}(\mathcal{F}_{\mathcal{R}})} \mathcal{P}(\omega) = 1.$$

Consequently, we have $\mathcal{P}(\phi) = 1$ and $\mathcal{R} \models_{\mathcal{P}} \phi$ holds. $\square$

Proposition 4.1.12 shows that probabilistic inferences defined as in Definition 4.1.10 enhance classical entailment in $\mathcal{RL}^S(\Sigma)$. We illustrate this by means of an example.

Example 4.1.13

We recall the knowledge base $\mathcal{R}_{\mathsf{bfp}}$ from Example 4.1.3(b), i.e., we consider $\mathcal{R}_{\mathsf{bfp}} = (\mathcal{F}_{\mathcal{R}_{\mathsf{bfp}}}, \mathcal{B}_{\mathcal{R}_{\mathsf{bfp}}})$ with

$$\begin{aligned} \mathcal{F}_{\mathcal{R}_{\mathsf{bfp}}} &= \{\forall X.(P(X) \Rightarrow B(X)),\ P(t)\}, \\ \mathcal{B}_{\mathcal{R}_{\mathsf{bfp}}} &= \{(F(X)|B(X))[0.9],\ (\overline{F(X)}|P(X))[0.99]\}. \end{aligned}$$

Further, let $\Sigma_{\mathsf{bfp}} = (\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}, \mathsf{Pred}_{\Sigma_{\mathsf{bfp}}})$ be an arbitrary signature with

$$t \in \mathsf{Const}_{\Sigma_{\mathsf{bfp}}}, \qquad \mathsf{Pred}_{\Sigma_{\mathsf{bfp}}} = \{B/1,\ F/1,\ P/1\},$$

and $|\mathsf{Const}_{\Sigma_{\mathsf{bfp}}}|$ large enough such that there is a probabilistic Σ_{bfp}-model of $\mathcal{R}_{\mathsf{bfp}}$

(the probability values of the conditionals in $\mathcal{R}_{\mathsf{bfp}}$ require a minimal number of constants). Then, for each such model $\mathcal{P} \in \mathsf{probMod}_{\Sigma_{\mathsf{bfp}}}(\mathcal{R}_{\mathsf{bfp}})$, we have $\mathcal{R}_{\mathsf{bfp}} \models_{\mathcal{P}} B(t)$, because

$$\mathcal{F}_{\mathcal{R}_{\mathsf{bfp}}} = \{\forall X.(P(X) \Rightarrow B(X)),\ P(t)\} \models_{\Sigma_{\mathsf{bfp}}} B(t).$$

Some "truly probabilistic" inferences $\mathcal{R} \models_{\mathcal{P}} (\psi|\phi)_{\mathsf{CS}}[p]$ with $p \in (0,1)$ might hold in all probabilistic models $\mathcal{P}$ of $\mathcal{R}$, too. See the next example for instance.

Example 4.1.14

1. Let $\Sigma_{\mathsf{opt}} = \{\emptyset, \{A/0, B/0\}\}$ and $\mathcal{R}_{\mathsf{opt}} = (\emptyset, \{(B|A)[p]\})$ with $p \in (0,1)$. Then, $\mathcal{R}_{\mathsf{opt}} \models_{\mathcal{P}} (\overline{B}|A)[1-p]$ for all Σ_{opt}-models $\mathcal{P}$ of $\mathcal{R}_{\mathsf{opt}}$ because $\mathcal{P}(A) > 0$ follows from $\mathcal{P} \models (B|A)[p]$ and

$$\begin{aligned}\mathcal{P}(\overline{B}|A) &= \frac{\mathcal{P}(A\overline{B})}{\mathcal{P}(A)} = \frac{\mathcal{P}(A) - \mathcal{P}(AB)}{\mathcal{P}(A)} \\ &= 1 - \frac{\mathcal{P}(AB)}{\mathcal{P}(A)} = 1 - \mathcal{P}(B|A) = 1 - p.\end{aligned}$$

2. We continue Example 4.1.13. Because of

$$\mathcal{P} \models \forall X.P(X) \Rightarrow B(X) \quad \text{and} \quad (\overline{F(X)}|P(X))[0.99],$$

one has

$$\mathcal{P} \models (B(X) \wedge \overline{F(X)}|P(X))[0.99].$$

This is because $\forall X.P(X) \Rightarrow B(X)$ states that every penguin is a bird and the conditional $(F(X)|P(X))[0.99]$ is concerned about penguins. Hence, in every possible world in which a ground instance $(F(t)|P(t))[0.99]$ of $(F(X)|P(X))[0.99]$ is applicable, the individual t is a penguin and, consequently, a bird. Adding this information to the consequence of the conditional does not change the set of verifying resp. falsifying possible worlds. Analogously, we can infer

$$\mathcal{P} \models (\overline{F(X)}|P(X) \wedge B(X))[0.99]$$

in all models $\mathcal{P}$ of $\mathcal{R}_{\mathsf{bfp}}$.

Because the inferences in the above-mentioned examples hold in all models $\mathcal{P}$ of the respective knowledge base $\mathcal{R}$, they could also be inferred from $\mathcal{R}$ with respect to the following skeptical inference relation that builds on all models of $\mathcal{R}$:

$$\mathcal{R} \models_{\Sigma} r \quad \text{iff} \quad \forall \mathcal{P} \in \mathsf{probMod}_{\Sigma}(\mathcal{R})\colon \mathcal{P} \models r.$$

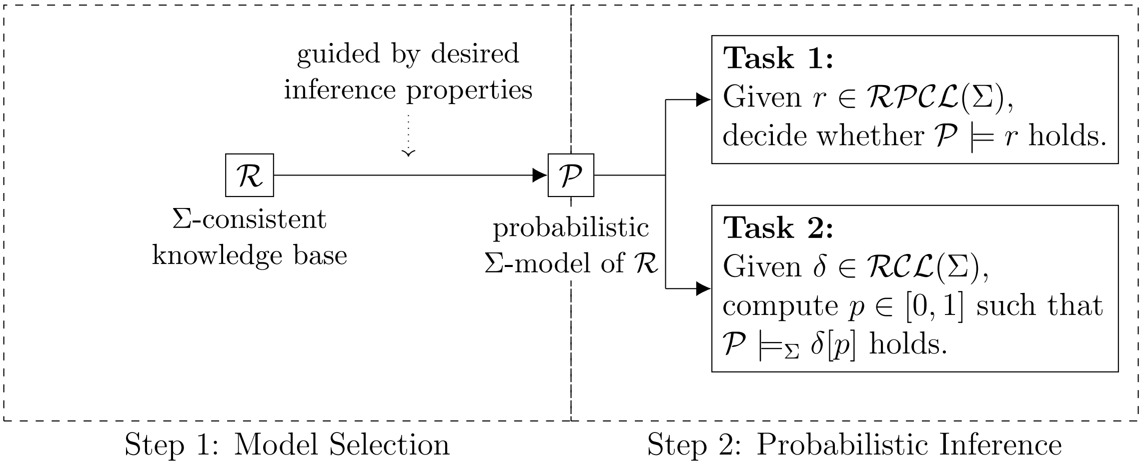

Figure 4.1: Answering probabilistic inferences.

The benefit of the skeptical inference relation $\models_\Sigma$ is that it does not require the selection of a specific model of $\mathcal{R}$. However, it is often considered as too cautious because it allows for few and little informative inferences only [Wilhelm et al., 2022]. If a single probabilistic model $\mathcal{P}$ of $\mathcal{R}$ is fixed, then it is possible that further inferences can be drawn which do not hold in all models of $\mathcal{R}$.

Definition 4.1.15: Model Selection Task *(cf. e.g., [Hastie et al., 2001])*

Let Σ be a finite signature, and let $\mathcal{R} \in \mathfrak{R}(\Sigma)$ be a Σ-consistent knowledge base. Then, the *model selection task* is the task of selecting a probabilistic Σ-model $\mathcal{P}$ of $\mathcal{R}$ from $\mathsf{probMod}_\Sigma(\mathcal{R})$.

The quality of the model selection is usually judged based on the quality of the inferences that can be drawn. It will turn out that the *maximum entropy model* (cf. Section 4.2) produces inferences of a very high quality in terms of inference properties. Based on the inference relation $\models_\mathcal{P}$ from Definition 4.1.10, we can conceptualize two *probabilistic inference tasks*.

Definition 4.1.16: Probabilistic Inference Task

Let Σ be a finite signature, and let $\mathcal{P}$ be a probabilistic Σ-model of a consistent knowledge base $\mathcal{R} \in \mathfrak{R}(\Sigma)$.

Task 1 (Decision Problem): Decide whether a probabilistic conditional $r \in \mathcal{RPCL}(\Sigma)$ can be inferred from $\mathcal{R}$ with respect to $\mathcal{P}$. That is, decide whether $\mathcal{R} \models_\mathcal{P} r$ holds or not. We denote this *decision problem* with "$\mathcal{R} \models_\mathcal{P} r?$."

Task 2 (Computation Problem): Compute the probability $p \in [0,1]$ with which a qualitative conditional $\delta \in \mathcal{RCL}(\Sigma)$ with $\mathcal{P}(\omega) \cdot \mathsf{app}_{\Sigma,\delta}(\omega) > 0$ for some $\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$ can be inferred from $\mathcal{R}$ with respect to $\mathcal{P}$. That is, compute $p \in [0,1]$ such that $\mathcal{R} \models_\mathcal{P} \delta[p]$ holds. We denote this *computation problem* with "$\mathcal{R} \models_\mathcal{P} \delta[?]$."

Altogether, the probabilistic inductive inference tasks consist of two steps, the *model selection* (Step 1) and the actual drawing of *probabilistic inferences* (Step 2; cf. Figure 4.1). In Step 1, a probabilistic Σ-model $\mathcal{P}$ of $\mathcal{R}$ is computed. This step is an inductive inference step because a knowledge base, which usually does not determine the probability values of all possible worlds, is extended to a complete belief state in which the probabilities of all possible worlds are fixed. In Step 2, the query answering takes place. That is, either it is decided whether $\mathcal{P} \models r$ holds for a probabilistic query conditional $r \in \mathcal{RPCL}(\Sigma)$ or the probability $p \in [0,1]$ is computed for which $\mathcal{P} \models \delta[p]$ holds where δ is a given query conditional $\delta \in \mathcal{RCL}(\Sigma)$.

Note that the probabilistic inference tasks in Definition 4.1.16 are well-defined in real-valued analysis. This is mainly because the Σ-consistency of $\mathcal{R}$ guarantees that a probabilistic Σ-model $\mathcal{P}$ of $\mathcal{R}$ exists. Further, in course of the computation problem, the claim that the query conditional δ satisfies $\mathcal{P}(\omega) \cdot \mathsf{app}_{\Sigma,\delta}(\omega) > 0$ for some $\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$ guarantees the well-definedness of the aggregating semantics (cf. Condition (3.11) of the characterization of the aggregating semantics). Note that the claim $\mathcal{P}(\omega) \cdot \mathsf{app}_{\Sigma,\delta}(\omega) > 0$ will reduce to the requirement that δ is applicable in at least one possible world, i.e., $\mathsf{app}_{\Sigma,\delta}(\omega) > 0$ for some $\omega \in \Omega(\Sigma)$, in the context of maximum entropy reasoning.

When calculating probabilistic inferences on a computer, one has to ensure that the probabilities occurring in the probabilistic inference tasks are rational numbers, though. Therefore, we will claim that the probabilities p of conditionals $r = \delta[p]$ in $\mathcal{R}$ and of the query conditional are rational numbers. But even then it is not guaranteed that the probabilistic model of $\mathcal{R}$ which shall be selected in the model selection task mentions rational probabilities only. Thus, we will inevitably have to deal with the approximation of probabilistic models.

In the next section, we will approach the model selection task (Step 1 of the probabilistic inference task) by following the *principle of maximum entropy.*

4.2 Principle of Maximum Entropy

The *principle of maximum entropy* [Jaynes, 1957a,b] constitutes a powerful methodology in probability theory which has been applied to *thermodynamics* [Dugdale, 1996; Frigg and Werndl, 2011] first and then found its way to many further disciplines including *information theory* [Shannon and Weaver, 1949; MacKay, 2003] and *knowledge representation and reasoning* (KR) [Paris and Vencovská, 1990; Paris, 1998, 2006]. In the context of KR it is assigned to the subfield of *Bayesian statistics* [Lee, 2012; Sivia and Skilling, 2006] where probabilities are understood as degrees of belief in the state of the real world. In this context, the principle of maximum entropy states that given some prior information about the world, here encoded in form of a knowledge base $\mathcal{R}$, the probability distribution which describes the state of the world best is the model of $\mathcal{R}$ which maximizes the *entropy*

$$\mathcal{H}(\mathcal{P}) = - \sum_{\omega \in \Omega(\Sigma)} \mathcal{P}(\omega) \cdot \log(\mathcal{P}(\omega)) \tag{4.6}$$

among all probabilistic Σ-models of $\mathcal{R}$. The reason for this opinion is that—from an information theoretical point of view—the model which maximizes the entropy makes as few additional assumptions as possible when completing the information in $\mathcal{R}$. In this sense, the principle of maximum entropy can be seen as an application of *Occam's razor* to probabilistic inductive reasoning (cf. [MacKay, 2003] for a discussion of Occam's razor in the context of probability theory). Also from a commonsense point of view, the *maximum entropy model* is considered to be the preferred choice for the model selection task. In [Paris, 1998] it is shown that in the propositional setting the maximum entropy model is the unique probability distribution satisfying some fundamental commonsense reasoning principles.

In this section, we formally define the maximum entropy model for relational probabilistic conditional knowledge bases, discuss the connection between the maximum entropy model and the so-called Σ-*p-consistency*, an accentuation of Σ-consistency, and apply the principle of maximum entropy to the probabilistic inductive inference task from Definition 4.1.16.

▶ Maximum Entropy Model

We discuss the maximum entropy model as an outstanding probabilistic model of a consistent knowledge base. Mathematically, as being the probabilistic model which maximizes the entropy (4.6), it is the solution of a convex optimization problem (cf. Proposition 4.2.12).

Definition 4.2.1: Maximum Entropy Model *(cf. Kern-Isberner [2004])*

Let Σ be a finite signature, and let $\mathcal{R} \in \mathfrak{R}(\Sigma)$ be a Σ-consistent knowledge base. Then, the *maximum entropy Σ-model* $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}} \colon \Omega(\Sigma) \to [0,1]$ of $\mathcal{R}$ is defined by

$$\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}} = \arg \max_{\mathcal{P} \in \mathsf{probMod}_{\Sigma}(\mathcal{R})} - \sum_{\omega \in \Omega(\Sigma)} \mathcal{P}(\omega) \cdot \log(\mathcal{P}(\omega)) \tag{4.7}$$

where the convention $0 \cdot \log(0) = 0$ applies. We call the maximization task in (4.7) the *(primary) maximum entropy optimization problem.*

If and only if the knowledge base $\mathcal{R}$ is Σ-consistent, then the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ exists and is unique (cf. e.g., [Boyd and Vandenberghe, 2009]). Hence, the Σ-consistency of $\mathcal{R}$ can be decided by finding out whether $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ exists or not.

By definition, for all probabilistic Σ-models of $\mathcal{R}$, we have $\mathcal{P}(\omega) = 0$ for all possible worlds $\omega \in \Omega^{0}_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$, i.e., for the possible worlds which violate any fact from $\mathcal{R}$ (cf. (4.4)). Consequently, this is true for the maximum entropy model as well, and we can restrict the sum in (4.7) to the possible worlds in $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$:

$$\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}} = \arg \max_{\mathcal{P} \in \mathsf{probMod}_{\Sigma}(\mathcal{R})} - \sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}(\omega) \cdot \log(\mathcal{P}(\omega)).$$

In the absence of uncertain beliefs, i.e., if the knowledge base $\mathcal{R}$ is of the form $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \emptyset)$, the maximum entropy Σ-model of $\mathcal{R}$ coincides with the *uniform distribution on* $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$. More precisely, $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}} = \mathcal{P}^{u}_{\Sigma,\mathcal{R}}$ where

$$\mathcal{P}^{u}_{\Sigma,\mathcal{R}}(\omega) = \begin{cases} |\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)|^{-1} & \text{if} \quad \omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma) \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

holds in this case. In particular, all potentially positive possible worlds are classified as equally probable. Note that $|\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)| > 0$ holds because if $\mathcal{R}$ is Σ-consistent, then there must be at least one possible world $\omega \in \Omega(\Sigma)$ which satisfies all the facts in $\mathcal{F}_{\mathcal{R}}$ by definition. We prove the statement formally.

Proposition 4.2.2: Maximum Entropy and the Uniform Distribution

Let Σ be a finite signature, and let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a Σ-consistent knowledge base with empty set of probabilistic beliefs, i.e., $\mathcal{B}_{\mathcal{R}} = \emptyset$. Then, the maximum entropy Σ-model of $\mathcal{R}$ is the uniform distribution on $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$, i.e., (cf. (4.8))

$$\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) = \mathcal{P}^{u}_{\Sigma,\mathcal{R}}(\omega), \qquad \omega \in \Omega(\Sigma),$$

and the entropy of $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ is

$$\mathcal{H}(\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}) = \mathcal{H}(\mathcal{P}^{u}_{\Sigma,\mathcal{R}}) = \log(|\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)|).$$

Proof. We apply the well-known method of Lagrangian multipliers (cf. e.g., [Bertsekas, 2014]) to the maximum entropy optimization problem in (4.7) and obtain the Lagrangian function

$$\begin{aligned}\Lambda &= -\sum_{\omega\in\Omega(\Sigma)} \mathcal{P}(\omega)\cdot\log(\mathcal{P}(\omega)) + \lambda\cdot(\sum_{\omega\in\Omega(\Sigma)} \mathcal{P}(\omega) - 1)\\ &= -\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}(\omega)\cdot\log(\mathcal{P}(\omega)) + \lambda\cdot(\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}(\omega) - 1)\end{aligned}$$

where Λ depends on the probabilities $\mathcal{P}(\omega)$ for $\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ and the Lagrangian multiplier λ which is introduced for the side condition $\sum_{\omega\in\Omega(\Sigma)}\mathcal{P}(\omega) = 1$. Note that we have applied $\mathcal{P}(\omega) = 0$ for all $\omega\in\Omega^{0}_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ to obtain the second formulation of Λ. The partial derivative of Λ with respect to the probability $\mathcal{P}(\omega)$ of an arbitrary potentially positive world $\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ is

$$\frac{\partial\Lambda}{\partial\mathcal{P}(\omega)} = -\log(\mathcal{P}(\omega)) - 1 + \lambda$$

which vanishes in $\mathcal{P}(\omega) = \exp(\lambda-1)$. Because $\mathcal{P}(\omega) = \exp(\lambda-1)$ has the same value for all $\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$, the optimum is reached for the probability distribution $\mathcal{P}$ with $\mathcal{P}(\omega) = \mathcal{P}(\omega')$ for all $\omega,\omega'\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ which indeed is the uniform distribution $\mathcal{P}^{u}_{\Sigma,\mathcal{R}}$. The optimum is a maximum because the Hessian matrix is negative definite as

$$\frac{\partial^2\Lambda}{\partial\mathcal{P}^{u}_{\Sigma,\mathcal{R}}(\omega)\partial\mathcal{P}^{u}_{\Sigma,\mathcal{R}}(\omega')} = \begin{cases}-\mathcal{P}^{u}_{\Sigma,\mathcal{R}}(\omega)^{-1} & \text{if}\quad \omega=\omega'\\ 0 & \text{otherwise}\end{cases}$$

and $\mathcal{P}^{u}_{\Sigma,\mathcal{R}}(\omega) > 0$ for all $\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ hold. Further, we have

$$\begin{aligned}\mathcal{H}(\mathcal{P}^{u}_{\Sigma,\mathcal{R}}) &= -\sum_{\omega\in\Omega(\Sigma)} \mathcal{P}^{u}_{\Sigma,\mathcal{R}}(\omega)\cdot\log(\mathcal{P}^{u}_{\Sigma,\mathcal{R}}(\omega))\\ &= -\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}^{u}_{\Sigma,\mathcal{R}}(\omega)\cdot\log(\mathcal{P}^{u}_{\Sigma,\mathcal{R}}(\omega))\\ &= -\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \frac{1}{|\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)|}\cdot\log(\frac{1}{|\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)|})\\ &= -|\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)|\cdot\frac{1}{|\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)|}\cdot\log(\frac{1}{|\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)|})\\ &= \log(|\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)|).\end{aligned}$$

The last equality holds because $-\log(\frac{1}{x}) = \log(x)$ for $x > 0$. □

As a consequence, we have the following estimation of the entropy of models of a consistent knowledge base $\mathcal{R}$.

Proposition 4.2.3: Upper and Lower Bounds of Entropy
Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a Σ-consistent knowledge base, and let $\mathcal{P}$ be a probabilistic Σ-model of $\mathcal{R}$. Then,

$$0 \leq \mathcal{H}(\mathcal{P}) \leq \log(|\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)|),$$

where $\mathcal{H}(\mathcal{P}) = 0$ holds if and only if $\mathcal{P}(\omega) \in \{0, 1\}$ for all $\omega \in \Omega(\Sigma)$.

Proof. Note that $\log(\mathcal{P}(\omega)) < 0$ holds for all $\mathcal{P}(\omega) \in (0, 1)$, and $\log(\mathcal{P}(\omega)) = 0$ holds if and only if $\mathcal{P}(\omega) = 1$. Further, recall that the convention $0 \cdot \log(0) = 0$ applies. Therewith, we have

$$-\mathcal{P}(\omega) \cdot \log \mathcal{P}(\omega) \begin{cases} = 0 & \text{if} \quad \mathcal{P}(\omega) \in \{0, 1\} \\ > 0 & \text{otherwise} \end{cases},$$

which proves both the non-negativity of $\mathcal{H}(\mathcal{P})$ and the fact that $\mathcal{H}(\mathcal{P}) = 0$ holds if and only if $\mathcal{P}(\omega) \in \{0, 1\}$ for all $\omega \in \Omega(\Sigma)$. In order to prove the upper bound $\mathcal{H}(\mathcal{P}) \leq \log(|\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)|)$, we note that every probabilistic Σ-model of $\mathcal{R}$ is also a probabilistic Σ-model of $\mathcal{R}' = (\mathcal{F}_{\mathcal{R}}, \emptyset)$ and, in accordance with Proposition 4.2.2, $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}'} = \mathcal{P}^{u}_{\Sigma,\mathcal{R}'}$ holds, where $\mathcal{P}^{u}_{\Sigma,\mathcal{R}'}$ is the uniform distribution on $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$. Therewith,

$$\mathcal{H}(\mathcal{P}) \leq \mathcal{H}(\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}'}) = \mathcal{H}(\mathcal{P}^{u}_{\Sigma,\mathcal{R}'}) = \log(|\Omega_{\mathcal{F}}(\Sigma)|)$$

for all $\mathcal{P} \in \mathsf{probMod}_{\Sigma}(\mathcal{R})$. □

The principle of maximum entropy tends to assign the same probability to possible worlds as long as nothing speaks against it—not only in the absence of uncertain beliefs ($\mathcal{B}_{\mathcal{R}} = \emptyset$). We will formalize this observation in Proposition 4.3.5. This tendency is a consequence of the fact that the principle of maximum entropy fulfills the *paradigm of information economy* [Gärdenfors, 1988] and adds missing probability values in an information theoretically optimal way.

A further consequence of the paradigm of information economy is that the principle of maximum entropy tends to assign probability values to possible worlds which are in the interior of the interval $[0, 1]$ and avoids the probabilities 0 and 1 if possible. This behavior is known as the *principle of open-mindedness*. According to [Kwong, 2016], open-mindedness is the "willingness to take a novel viewpoint seriously." In the probabilistic context, open-mindedness can be understood in such a way that an open-minded person tends to not commit to factual probabilities (0 or 1). Also see [Paris, 2006] for a discussion of the open-mindedness principle in the context of maximum entropy reasoning.

We can say that the maximum entropy model of a consistent knowledge base $\mathcal{R}$ is the model of $\mathcal{R}$ which is as cautious as possible [Kern-Isberner, 2001a]. Nevertheless, there are some constellations in which the principle of maximum entropy assigns to potentially positive possible worlds from $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ 0-1-probabilities, too.

Example 4.2.4
We consider the signature $\Sigma_{\mathsf{ex}} = (\emptyset, \{A/0, B/0\})$ and the knowledge base

$$\mathcal{R}_{\mathsf{ex}} = (\emptyset, \{(AB|\top)[0.1],\ (A\overline{B}|\top)[0.3],\ (\overline{A}B|\top)[0.6]\}).$$

$\mathcal{R}_{\mathsf{ex}}$ is Σ_{ex}-consistent, where the probability distribution $\mathcal{P}_{\mathsf{ex}}$ on $\Omega(\Sigma_{\mathsf{ex}})$ given by

$$\begin{aligned} \mathcal{P}_{\mathsf{ex}}(AB) &= 0.1, & \mathcal{P}_{\mathsf{ex}}(A\overline{B}) &= 0.3, \\ \mathcal{P}_{\mathsf{ex}}(\overline{A}B) &= 0.6, & \mathcal{P}_{\mathsf{ex}}(\overline{A}\overline{B}) &= 0, \end{aligned}$$

is the only Σ_{ex}-model of $\mathcal{R}_{\mathsf{ex}}$. The first three probability assignments can be read from the knowledge base directly. Further, because of the normalization condition of probability distributions, $\mathcal{P}_{\mathsf{ex}}$ has to satisfy

$$\begin{aligned} \mathcal{P}_{\mathsf{ex}}(\overline{A}\overline{B}) &= 1 - (\mathcal{P}_{\mathsf{ex}}(AB) + \mathcal{P}_{\mathsf{ex}}(A\overline{B}) + \mathcal{P}_{\mathsf{ex}}(\overline{A}B)) \\ &= 1 - (0.1 + 0.3 + 0.6) = 0. \end{aligned}$$

That is, $\mathcal{P}_{\mathsf{ex}}(\overline{A}\overline{B}) = 0$ holds although $\overline{A}\overline{B} \in \Omega_{\mathcal{F}_{\mathcal{R}_{\mathsf{ex}}}}(\Sigma_{\mathsf{ex}})$. Because $\mathcal{P}_{\mathsf{ex}}$ is the only Σ_{ex}-model of $\mathcal{R}_{\mathsf{ex}}$, it has to be the maximum entropy Σ_{ex}-model of $\mathcal{R}_{\mathsf{ex}}$ which shows that, in some rare cases, maximum entropy models assign 0-1-probabilities to potentially positive possible worlds $\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$.

Some numerical methods for computing maximum entropy models can be applied to *positive* probability distributions only [Boyd and Vandenberghe, 2009], which means that $\mathcal{P}(\omega) > 0$ must hold for all $\omega \in \Omega(\Sigma)$, respectively for all $\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ if the probability distributions are restricted to the possibly positive possible worlds in $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$. Example 4.2.4 shows that the Σ-consistency of $\mathcal{R}$ is not sufficient to ensure the existence of such a *positive* Σ*-model* of $\mathcal{R}$ or the positivity of the maximum entropy Σ-model. Thus, from now on, we will focus on knowledge bases which satisfy a slightly stronger version of consistency that is called Σ*-p-consistency* and which guarantees the existence of positive models (cf. [Finthammer, 2017]).

▶ Maximum Entropy and Σ-p-Consistency

The Σ-*p-consistency* of a knowledge base $\mathcal{R}$ ensures that the knowledge base does not only have a Σ-model but also one with solely *positive* values (cf. [Finthammer, 2017]). Here, we require the positivity on the subset $\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$ only, which is sufficient because the computation of probabilistic models over $\Omega(\Sigma)$ can be reduced to the computation of models over $\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$ as discussed above. In other words, we assume that possible worlds $\omega \in \Omega^0_{\mathcal{F}_\mathcal{R}}(\Sigma)$, for which $\mathcal{P}(\omega) = 0$ holds for all models $\mathcal{P}$ of $\mathcal{R}$ by definition, are filtered out in a preprocessing step if necessary.

Definition 4.2.5: $\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$-Positive Σ-Model *(cf. [Finthammer, 2017])*

Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_\mathcal{R}, \mathcal{B}_\mathcal{R})$ be a Σ-consistent knowledge base, and let $\mathcal{P}$ be a Σ-model of $\mathcal{R}$. Then, $\mathcal{P}$ is called $\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$*-positive* if $\mathcal{P}(\omega) > 0$ holds for all $\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$.

In $\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$-positive Σ-models, the possibly positive possible worlds indeed are positive. As said, the Σ-p-consistency of a knowledge base ensures the existence of such a model.

Definition 4.2.6: Σ-p-Consistency *(cf. [Finthammer, 2017])*

Let Σ be a finite signature, and let $\mathcal{R} = (\mathcal{F}_\mathcal{R}, \mathcal{B}_\mathcal{R})$ be a knowledge base. Then, $\mathcal{R}$ is called Σ-*p-consistent* if there is an $\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$*-positive* Σ-*model* of $\mathcal{R}$.

Demanding that a knowledge base is not only Σ-consistent but also Σ-p-consistent is not a strong restriction but rather a quality criterion for good knowledge engineering. Knowledge bases $\mathcal{R}$ which are Σ-consistent but not Σ-p-consistent are of such a form that there is a possibly positive world $\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$ with $\mathcal{P}(\omega) = 0$ in all probabilistic Σ-models $\mathcal{P} \in \mathsf{probMod}_\Sigma(\mathcal{R})$ (cf. Example 4.2.4). This means that factual knowledge is expressed by conditional beliefs in $\mathcal{B}_\mathcal{R}$ instead of sentences in $\mathcal{F}_\mathcal{R}$.

By introducing additional facts in $\mathcal{F}_\mathcal{R}$, every Σ-consistent knowledge base $\mathcal{R}$ can be transformed into a Σ-p-consistent knowledge base $\mathcal{R}'$ with $\mathsf{probMod}_\Sigma(\mathcal{R}') = \mathsf{probMod}_\Sigma(\mathcal{R})$. This is possible because positive Σ-models have to be positive on $\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$ only. The idea of the transformation is to introduce for every potentially positive world $\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$ which is forced to have the probability $\mathcal{P}(\omega) = 0$ in every probabilistic Σ-model $\mathcal{P}$ of $\mathcal{R}$ a fresh sentence $F \in \mathcal{RL}^S(\Sigma)$ in the set of facts $\mathcal{F}_\mathcal{R}$ with $F = \neg\omega$. Then, $\mathcal{P}(\omega) = 0$ holds because of the fact F already, and ω no longer causes a violation of the positivity condition of models of $\mathcal{R}$ as ω is an element of $\Omega^0_{\mathcal{F}_\mathcal{R}}(\Sigma)$ now. We illustrate this idea by means of an example.

Example 4.2.7
We recall the knowledge base

$$\mathcal{R}_{\mathsf{ex}} = (\emptyset, \{(AB|\top)[0.1],\ (A\overline{B}|\top)[0.3],\ (\overline{A}B|\top)[0.6]\})$$

from Example 4.2.4. As discussed in Example 4.2.4, the conditionals in $\mathcal{B}_{\mathcal{R}_{\mathsf{ex}}}$ force the unique probabilistic Σ_{ex}-model $\mathcal{P}_{\mathsf{ex}}$ of $\mathcal{R}_{\mathsf{ex}}$ to satisfy $\mathcal{P}_{\mathsf{ex}}(\overline{A}\overline{B}) = 0$. Hence, $\overline{A}\overline{B}$ is a potentially positive possible world with probability 0 which violates the positivity condition of Σ-p-consistent knowledge bases. This problem can be prevented by transforming $\mathcal{R}_{\mathsf{ex}}$ into the knowledge base

$$\mathcal{R}'_{\mathsf{ex}} = (\mathcal{F}'_{\mathcal{R}_{\mathsf{ex}}}, \mathcal{B}_{\mathcal{R}_{\mathsf{ex}}}) = (\{\neg(\overline{A}\overline{B})\}, \{(AB|\top)[0.1],\ (A\overline{B}|\top)[0.3],\ (\overline{A}B|\top)[0.6]\}).$$

Then, $\mathcal{P}_{\mathsf{ex}}$ is still a model of $\mathcal{R}'_{\mathsf{ex}}$ and $\mathsf{probMod}_{\Sigma_{\mathsf{ex}}}(\mathcal{R}_{\mathsf{ex}}) = \mathsf{probMod}_{\Sigma_{\mathsf{ex}}}(\mathcal{R}'_{\mathsf{ex}})$ holds but $\overline{A}\overline{B}$ is no longer a potentially positive possible world but an element of $\Omega^0_{\mathcal{F}'_{\mathcal{R}_{\mathsf{ex}}}}(\Sigma)$ now. Thus, $\mathcal{P}(\omega) > 0$ holds for all $\omega \in \Omega_{\mathcal{F}'_{\mathcal{R}_{\mathsf{ex}}}}(\Sigma)$.

We give a further example of a Σ-p-consistent knowledge base.

Example 4.2.8
The knowledge base $\mathcal{R}_{\mathsf{bow}} = (\emptyset, \{r_1, r_2\})$ from Example 4.1.3(a) with

$$r_1 = (B|SR)[0.6] \quad \text{and} \quad r_2 = (\overline{B}|\overline{S} \vee \overline{R})[0.99]$$

defined over $\Sigma_{\mathsf{bow}} = (\emptyset, \mathsf{Pred}_{\mathsf{bow}})$ is Σ_{bow}-p-consistent. To see this, let $\mathcal{P}_{\mathsf{bow}}$ be the probability distribution over $\Omega(\Sigma_{\mathsf{bow}})$ with

$$\begin{aligned}
\mathcal{P}_{\mathsf{bow}}(BSR) &= 0.15,\\
\mathcal{P}_{\mathsf{bow}}(\overline{B}SR) &= 0.1,\\
\mathcal{P}_{\mathsf{bow}}(\overline{B}\,\overline{S}R) = \mathcal{P}_{\mathsf{bow}}(\overline{B}S\overline{R}) = \mathcal{P}_{\mathsf{bow}}(\overline{B}\,\overline{S}\,\overline{R}) &= 0.2475,\\
\mathcal{P}_{\mathsf{bow}}(B\overline{S}R) = \mathcal{P}_{\mathsf{bow}}(BS\overline{R}) = \mathcal{P}_{\mathsf{bow}}(B\overline{S}\,\overline{R}) &= 0.0025.
\end{aligned}$$

Then, $\mathcal{P}_{\mathsf{bow}}$ is positive and a model of $\mathcal{R}_{\mathsf{bow}}$ because

$$\mathcal{P}_{\mathsf{bow}}(B|SR) = \frac{\mathcal{P}_{\mathsf{bow}}(BSR)}{\mathcal{P}_{\mathsf{bow}}(BSR) + \mathcal{P}_{\mathsf{bow}}(\overline{B}SR)} = \frac{0.15}{0.15 + 0.1} = 0.6,$$

$$\mathcal{P}_{\mathsf{bow}}(\overline{B}|\overline{S} \vee \overline{R}) = \frac{\mathcal{P}_{\mathsf{bow}}(\overline{B}\,\overline{S}R) + \mathcal{P}_{\mathsf{bow}}(\overline{B}S\overline{R}) + \mathcal{P}_{\mathsf{bow}}(\overline{B}\,\overline{S}\,\overline{R})}{1 - (\mathcal{P}_{\mathsf{bow}}(BSR) + \mathcal{P}_{\mathsf{bow}}(\overline{B}SR))}$$

$$= \frac{3 \cdot 0.2475}{1 - (0.15 + 0.1)} = 0.99.$$

The Σ-p-consistency of a knowledge base $\mathcal{R}$ has important consequences on the computation of the maximum entropy model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$. In general, not all Σ-models of a Σ-p-consistent knowledge base $\mathcal{R}$ are positive on $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$, but, importantly, this holds for the maximum entropy model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ (cf. [Boyd and Vandenberghe, 2009]). This is due to the fact that the maximum entropy optimization problem is a convex optimization problem which again is, above all, a consequence of the linearity of the acceptance condition of conditionals as proposed by the aggregating semantics (cf. the characterization of the aggregating semantics in Proposition 3.4.5 and Proposition 4.2.12 on the convexity of the maximum entropy optimization problem). Thus, the following proposition, which is known from the propositional case, can be "lifted" to the case where the conditionals in $\mathcal{R}$ are relational.

Proposition 4.2.9: Positivity of Maximum Entropy Model

(cf. [Boyd and Vandenberghe, 2009])

Let Σ be a finite signature, and let $\mathcal{R} \in \mathfrak{R}(\Sigma)$ be a Σ-p-consistent knowledge base. Then, $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ is positive on $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$.

Proof Idea. The idea of the proof is that the set $\mathsf{probMod}_{\Sigma}(\mathcal{R})$ of all Σ-models of $\mathcal{R}$ defines a convex subset $\mathcal{S} \subseteq [0,1]^{|\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)|}$ and that the Σ-p-consistency of $\mathcal{R}$ guarantees the existence of a point P in the relative interior of $\mathcal{S}$. Approaching the boundary of $\mathcal{S}$ from this inner point, i.e., approaching the probability $\mathcal{P}(\omega) = 0$ (or $\mathcal{P}(\omega) = 1$) for some $\omega \in \mathcal{P}(\omega)$ on any straight line through P decreases the entropy of the respective model. See [Boyd and Vandenberghe, 2009] for the technical details. $\square$

As a consequence, we obtain the following property of the maximum entropy Σ-model which underpins the fact that it does not generate 0-probabilities by chance.

Proposition 4.2.10: Probabilistic Classicality

Let Σ be a finite signature, and let $\mathcal{R}$ be a Σ-p-consistent knowledge base. Then, for $\omega \in \Omega(\Sigma)$, it holds that $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) = 0$ if and only if $\mathcal{P}(\omega) = 0$ for all $\mathcal{P} \in \mathsf{probMod}_{\Sigma}(\mathcal{R})$.

Proof. Let $\omega \in \Omega(\Sigma)$, and let $\mathcal{P}(\omega) = 0$ for all $\mathcal{P} \in \mathsf{probMod}_{\Sigma}(\mathcal{R})$. Then, $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}} = 0$ follows because $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}} \in \mathsf{probMod}_{\Sigma}(\mathcal{R})$. Now, let $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) = 0$ for $\omega \in \Omega(\Sigma)$. Then, $\omega \in \Omega^{0}_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ according to Proposition 4.2.9. Consequently, $\mathcal{P}(\omega) = 0$ for all $\mathsf{probMod}_{\Sigma}(\mathcal{R})$ according to (4.4). $\square$

A further consequence is that we can reduce the search space of the maximum entropy optimization problem in (4.7) to models of $\mathcal{R}$ which are positive on $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$. Then, the condition

$$\sum_{\omega \in \Omega(\Sigma)} \mathsf{app}_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) > 0 \tag{4.9}$$

from the characterization of the aggregating semantics (cf. Proposition 3.4.5) becomes vacuous because it is satisfied by every $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$-positive Σ-model of $\mathcal{R}$, as the following proposition shows.

Proposition 4.2.11: Impact of Positivity on Aggregating Semantics
Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a Σ-p-consistent knowledge base, let $r \in \mathcal{B}_{\mathcal{R}}$, and let $\mathcal{P}$ be a probability distribution which is positive on $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$. Then,

$$\sum_{\omega \in \Omega(\Sigma)} \mathsf{app}_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) > 0.$$

In particular, this holds for all $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$-positive Σ-models of $\mathcal{R}$ including the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ of $\mathcal{R}$.

Proof. Because $\mathcal{R}$ is Σ-consistent, there is a Σ-model $\mathcal{P}'$ of $\mathcal{R}$. According to the aggregating semantics,

$$\sum_{\omega \in \Omega(\Sigma)} \mathsf{app}_{\Sigma,r}(\omega) \cdot \mathcal{P}'(\omega) > 0$$

for all $r \in \mathcal{B}_{\mathcal{R}}$ holds. Further, because $\mathsf{app}_{\Sigma,r}(\omega) \geq 0$ for all $\omega \in \Omega(\Sigma)$ and $\mathcal{P}'(\omega) = 0$ for all $\omega \in \Omega^0_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$, we have

$$\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathsf{app}_{\Sigma,r}(\omega) \cdot \mathcal{P}'(\omega) > 0.$$

Thus, there is at least one possible world $\omega' \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ with $\mathsf{app}_{\Sigma,r}(\omega') \cdot \mathcal{P}'(\omega') > 0$. Particularly, $\mathsf{app}_{\Sigma,r}(\omega') > 0$ holds. Now, let $\mathcal{P}$ be a probability distribution which is positive on $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$. Then, $\mathcal{P}(\omega') > 0$ holds and, therewith, $\mathsf{app}_{\Sigma,r}(\omega') \cdot \mathcal{P}(\omega') > 0$ follows. Consequently, we have

$$\sum_{\omega \in \Omega(\Sigma)} \mathsf{app}_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) > 0$$

which proves the proposition. □

When dropping the condition (4.9) from the characterization of the aggregating semantics, we obtain the following characterization of the maximum entropy optimization problem in (4.7) which is a convex optimization problem in standard form according to [Boyd and Vandenberghe, 2009].

Proposition 4.2.12: Maximum Entropy Optimization Problem
Let Σ be a finite signature, and let $\mathcal{R}$ be a Σ-p-consistent knowledge base. Then, the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ is the unique solution of the convex optimization problem

$$\begin{aligned} \text{minimize} \sum_{\omega \in \Omega(\Sigma)} \mathcal{P}(\omega) \cdot \log(\mathcal{P}(\omega)) \qquad & \qquad (4.10) \\ \text{subject to} \qquad & \\ \sum_{\omega \in \Omega(\Sigma)} f_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) = 0, \qquad & \forall r \in \mathcal{B}_{\mathcal{R}}, \\ \sum_{\omega \in \Omega(\Sigma)} \mathcal{P}(\omega) = 1, \qquad & \\ \mathcal{P}(\omega) = 0, \qquad & \forall \omega \in \Omega^0_{\mathcal{F}_{\mathcal{R}}}(\Sigma), \\ -\mathcal{P}(\omega) \leq 0, \qquad & \forall \omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma). \end{aligned}$$

The feature function $f_{\Sigma,r}(\omega)$ can be replaced in (4.10) by the standardized feature function $\hat{f}_{\Sigma,r}(\omega)$ (cf. Definition 3.4.10) one-to-one.

Proof. The optimization problem (4.10) is the maximum entropy optimization problem in (4.7) except for the condition (4.9) of the characterization of the aggregating semantics (cf. Proposition 3.4.5). In (4.10), the requirements that $\mathcal{P}$ has to be a probability distribution and it has to be a probabilistic Σ-model of $\mathcal{R}$ are made explicit. In detail, the normalization condition $\sum_{\omega \in \Omega(\Sigma)} \mathcal{P}(\omega) = 1$ and the non-negativity condition $-\mathcal{P}(\omega) \leq 0$ guarantee that $\mathcal{P}$ is a probability distribution over $\Omega(\Sigma)$, and $\sum_{\omega \in \Omega(\Sigma)} f_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) = 0$ for all $r \in \mathcal{B}_{\mathcal{R}}$ as well as $\mathcal{P}(\omega) = 0$ for all $\omega \in \Omega^0_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ guarantee that $\mathcal{P}$ is a Σ-model of $\mathcal{R}$. Here, the requirement $\sum_{\omega \in \Omega(\Sigma)} \mathsf{app}_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) > 0$ can be dropped because of the previous Proposition 4.2.11.

The optimization problem (4.10) is convex, because the negative entropy is a strictly convex function [Rao, 1984] and the side conditions of the optimization problem form a linear program, i.e., a system of linear (in-)equations over continuous variables. Hence, the solution space is a convex polytope (cf. [Schrijver, 1998]). □

Because every probabilistic model of a consistent knowledge base leads to a probabilistic inference relation (cf. Definition 4.1.10), this also holds for the maximum entropy model. We briefly discuss the resulting maximum entropy inference relation next.

▶ Maximum Entropy Inference

We make use of the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ of a Σ-p-consistent knowledge base $\mathcal{R}$ for drawing inferences. The maximum entropy inference relation $\models^{\mathsf{ME}}_{\Sigma}$ is the only model-based probabilistic inference relation which satisfies a variety of meaningful commonsense principles (see [Paris, 1998] for the details in the propositional case). Roughly said, the inferred maximum entropy probabilities are the most expected probability values when $\mathcal{R}$ is assumed to hold. Therewith, $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ constitutes a distinguished model for the model-based probabilistic inductive inference task showcased in Figure 4.1.

Definition 4.2.13: Maximum Entropy Inference Relation

(cf. Kern-Isberner and Thimm [2010])

Let Σ be a finite signature, let $\mathcal{R}$ be a Σ-p-consistent knowledge base, and let $r \in \mathcal{RPCL}(\Sigma)$ be a probabilistic conditional. Then, the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ of $\mathcal{R}$ yields the *maximum entropy inference relation*

$$\mathcal{R} \models^{\mathsf{ME}}_{\Sigma} r \quad \text{iff} \quad \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}} \models r. \tag{4.11}$$

Recall that the maximum entropy model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ is the solution of an optimization problem and can only be computed numerically in general. Consequently, maximum entropy inferences can usually be computed approximately only, too.

Example 4.2.14

We consider the signature $\Sigma_{\mathsf{bow}} = (\emptyset, \{B/0, S/0, R/0\})$ and the knowledge base

$$\mathcal{R}_{\mathsf{bow}} = (\emptyset, \{(B|SR)[0.6],\ (\overline{B}|\overline{S} \vee \overline{R})[0.99]\})$$

from Example 4.1.3. In Example 4.3.9, we will show that the maximum entropy Σ_{bow}-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}$ of $\mathcal{R}_{\mathsf{bow}}$ is given by

$$\begin{aligned}
\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(BSR) &\approx 0.229,\\
\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(BS\overline{R}) = \mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(B\overline{S}R) = \mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(B\overline{S}\overline{R}) &\approx 0.002,\\
\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(\overline{B}SR) &\approx 0.153,\\
\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(\overline{B}S\overline{R}) = \mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(\overline{B}\overline{S}R) = \mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(\overline{B}\overline{S}\overline{R}) &\approx 0.204,
\end{aligned}$$

where the probabilities are rounded to three decimal places. Alternatively, $\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}$ can be computed with the software tool Spirit [Meyer and

Rödder, 1996; Rödder and Meyer, 1996]. For example, because

$$\begin{aligned}
&\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(B|R) = \frac{\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(BR)}{\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(R)}\\
&= \frac{\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(BSR) + \mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(B\overline{S}R)}{\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(BSR) + \mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(BS\overline{R}) + \mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(B\overline{S}R) + \mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(\overline{B}\overline{S}R)}\\
&\approx \frac{0.229 + 0.002}{0.229 + 0.002 + 0.002 + 0.204} = \frac{0.231}{0.437} = 0.529,
\end{aligned}$$

we can (approximately) draw the maximum entropy inference

$$\mathcal{R}_{\mathsf{bow}} \models^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}}} (B|R)[0.529].$$

It is worth emphasizing that $\models^{\mathsf{ME}}_{\Sigma}$ is a nonmonotonic inference relation, i.e., $\mathcal{R} \models^{\mathsf{ME}}_{\Sigma} r$ and $\mathcal{R} \sqsubseteq \mathcal{R}'$ does not necessarily lead to $\mathcal{R}' \models^{\mathsf{ME}}_{\Sigma} r$. When reasoning about uncertain knowledge in flux, nonmonotonicity is a preferable property.

In the following section, we examine the dual maximum entropy problem, revealing how conditional structures impact maximum entropy models. Insights into the internal structure of these models can enhance our understanding of their composition and aid in designing numerical solvers more suited to the maximum entropy optimization.

4.3 Dual Maximum Entropy Problem

In mathematical optimization, the *principle of duality* (cf. e.g., [Boyd and Vandenberghe, 2009]) is an approach to solve an optimization problem by transforming it into its *dual optimization problem* beforehand. The dual optimization problem is obtained by applying the well known method of *Lagrangian multipliers* (cf. e.g., [Bertsekas, 2014]) to the primal problem in standard form. In some cases, the dual optimization problem has a much more compact representation than the primal problem and, moreover, it is unconstrained, which allows one to apply more efficient solving techniques to the dual problem than to the primal problem.

While the primal and its dual optimization problem do not yield the same optimum in general, there are specific conditions to the primal optimization problem under which the equality of the optima is guaranteed. In this case, one speaks of *strong duality* between the primal and the dual optimization problem. In the general case, the conditions for strong duality are known as the *Karush-Kuhn-Tucker conditions* [Kuhn, 1982; Kuhn and Tucker, 1951]. If the optimization problem is convex, then the Karush-Kuhn-Tucker conditions can be relaxed to *Slater's condition* [Slater, 1950]. Basically, Slater's condition demands the existence of an interior point in the feasible region, i.e., in the solution space of the optimization problem. Please see Definition A.1.6 in the appendix for a formal definition of Slater's condition.

In this section, we apply the principle of duality to the maximum entropy optimization problem in standard form (4.10). This is possible if the knowledge base $\mathcal{R}$ is Σ-p-consistent because Slater's condition is satisfied then. The dual maximum entropy optimization leads to a product representation of the maximum entropy model which we discuss in detail. Eventually, we present the *maximum entropy equation system* as a further starting point for maximum entropy computations.

▶ Transformation of the Maximum Entropy Problem

According to Proposition 4.2.12, the maximum entropy optimization problem (4.10) is convex and the Σ-p-consistency of the considered knowledge base $\mathcal{R}$ guarantees the existence of an interior point in the feasible region of (4.10) such that Slater's condition applies. Here, the importance of the Σ-p-consistency of knowledge bases becomes clear. As a consequence, we can consider the optimization problem dual to (4.10) which leads to the same optimum as the primal problem because of the strong duality of the optimization problems.

In order to apply the principle of duality, we consider a Σ-p-consistent knowledge base $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ and, without loss of generality, we enumerate the conditionals in $\mathcal{B}_{\mathcal{R}}$, i.e., we identify $\mathcal{B}_{\mathcal{R}}$ with an enumerated set of conditionals $\{r_1, \dots, r_n\}$ with $r_i = (\psi_i|\phi_i)_{\mathsf{CS}_i}[p_i]$ for $i = 1, \dots, n$ in the following.

Definition 4.3.1: Dual Maximum Entropy Optimization Problem
(cf. [Boyd and Vandenberghe, 2009; Kern-Isberner, 2001a])

Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a Σ-p-consistent knowledge base, and let $f_{\Sigma,i}$ for $i = 1, \dots, n$ be the feature functions of $r_i \in \mathcal{B}_{\mathcal{R}}$ (cf. Definition 3.4.3). Then, we call the (unconstrained) optimization problem

$$\begin{aligned} &\text{minimize} && \sum_{\omega \in \Omega_{\mathcal{F}}(\Sigma)} \prod_{i=1}^{n} x_i^{f_{\Sigma,i}(\omega)} && (4.12) \\ &\text{subject to} && x_i > 0, \qquad i = 1, \dots, n, \end{aligned}$$

the *dual maximum entropy optimization problem* of $\mathcal{R}$ with respect to Σ. A solution vector $\vec{\alpha}_{\Sigma,\mathcal{R}} = (\alpha_1, \dots, \alpha_n) \in \mathbb{R}^n_{>0}$ of (4.12) is called *maximum entropy vector* of $\mathcal{R}$ with respect to Σ, or Σ-ME-*vector* of $\mathcal{R}$ for short.

We will show soon that (4.12) is indeed dual to the primal maximum entropy optimization problem (4.10). Due to Proposition 3.4.9, the feature functions $f_{\Sigma,i}$ can be replaced in (4.12) by the standardized feature functions $\hat{f}_{\Sigma,i}$ (cf. Definition 3.4.10) one-to-one.

An obvious benefit of considering the dual maximum entropy optimization problem is that solution vectors $\vec{\alpha}_{\Sigma,\mathcal{R}}$ of (4.12) are of length $n = |\mathcal{B}_{\mathcal{R}}|$. That is, the number of values that have to be computed is constant in the domain size $k = |\mathsf{Const}_{\Sigma}|$ in contrast to the exponentially many probability values that have to be calculated in order to solve the primal problem (4.10) directly. From such a solution vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$, the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ can be derived as follows.

Proposition 4.3.2: Product Representation of Maximum Entropy Model *(cf. [Kern-Isberner, 2001a; Jaynes, 1983])*

Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a Σ-p-consistent knowledge base, let $f_{\Sigma,i}$ for $i = 1, \ldots, n$ be the feature functions of $r_i \in \mathcal{B}_{\mathcal{R}}$, and let $\vec{\alpha}_{\Sigma,\mathcal{R}} = (\alpha_1, \ldots, \alpha_n)$ be a Σ-ME-vector of $\mathcal{R}$, i.e., a solution of (4.12). Then, for the maximum entropy Σ-model of $\mathcal{R}$ it holds that

$$\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) = \begin{cases} \alpha_0 \cdot \prod_{i=1}^{n} \alpha_i^{f_{\Sigma,i}(\omega)} & \text{if} \quad \omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma) \\ 0 & \text{otherwise} \end{cases}, \tag{4.13}$$

where α_0 is a normalizing constant given by

$$\alpha_0 = \left(\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \prod_{i=1}^{n} \alpha_i^{f_{\Sigma,i}(\omega)} \right)^{-1}. \tag{4.14}$$

Due to Proposition 3.4.9, the feature functions $f_{\Sigma,i}$ can be replaced in (4.13) and (4.14) by the standardized feature functions $\hat{f}_{\Sigma,i}$ (cf. Definition 3.4.10) one-to-one.

Proof. We start from the primal maximum entropy optimization problem (4.10) and derive its dual problem by using the method of Lagrangian multipliers. Beforehand, we simplify the objective function $\sum_{\omega \in \Omega(\Sigma)} \mathcal{P}(\omega) \cdot \log(\mathcal{P}(\omega))$ and the constraints in (4.10) by exploiting the fact that $\mathcal{P}(\omega) = 0$ holds for $\omega \in \Omega^0_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$. That is, we build the sums in (4.10) over the potentially positive worlds only. Then, the Lagrangian function of the optimization problem (4.10) becomes (cf. [Boyd and Vandenberghe, 2009])

$$\Lambda(\mathcal{P}(\omega_1), \ldots, \mathcal{P}(\omega_k), \lambda_0, \lambda_1, \ldots, \lambda_n) = \sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}(\omega) \cdot \log(\mathcal{P}(\omega))$$

$$+ \lambda_0 \cdot (\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}(\omega) - 1) + \sum_{i=1}^{n} \lambda_i \cdot \sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\omega)} f_{\Sigma,i}(\omega) \cdot \mathcal{P}(\omega),$$

where $\lambda_0, \lambda_1, \ldots, \lambda_n$ are the Lagrangian multipliers and where $\omega_1, \ldots, \omega_k$ are the possible worlds in $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$. The dual problem of (4.10) is (cf. [Boyd and Vandenberghe, 2009])

$$\text{maximize} \quad g(\lambda_0, \lambda_1, \ldots, \lambda_n) = \inf\nolimits_{\mathcal{P}} \Lambda(\mathcal{P}(\omega_1), \ldots, \mathcal{P}(\omega_k), \lambda_0, \lambda_1, \ldots, \lambda_n). \tag{4.15}$$

We determine the derivation of Λ with respect to the probability of an arbitrary possible world $\omega' \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ and get

$$\frac{\partial \Lambda}{\partial \mathcal{P}(\omega')} = \log(\mathcal{P}(\omega')) + 1 + \lambda_0 + \sum_{i=1}^{n} \lambda_i \cdot f_{\Sigma,i}(\omega')$$

which vanishes (only) in

$$\mathcal{P}(\omega') = \exp(-1 - \lambda_0 - \sum_{i=1}^{n} \lambda_i \cdot f_{\Sigma,i}(\omega')). \tag{4.16}$$

Consequently, this determines the only candidate where Λ can take its infimum, which is actually a minimum, and the principle of strong duality tells us that the minimum is reached. We replace the log-term in the Lagrangian function by the right-hand side of (4.16) and obtain

$$\begin{aligned}
\inf\nolimits_{\mathcal{P}} \Lambda = & \sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}(\omega) \cdot (-1 - \lambda_0 - \sum_{i=1}^{n} \lambda_i \cdot f_{\Sigma,i}(\omega)) + \lambda_0 \cdot (\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}(\omega) - 1) \\
& + \sum_{i=1}^{n} \lambda_i \cdot \sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\omega)} f_{\Sigma,i}(\omega) \cdot \mathcal{P}(\omega) \\
= & - \sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}(\omega) - \lambda_0 \\
= & -1 - \lambda_0.
\end{aligned}$$

Afterwards, we plug this result in (4.15) and obtain that the dual problem becomes maximizing $-1 - \lambda_0$. Instead of considering this maximization problem directly, we substitute $x_0 = \exp(-1 - \lambda_0)$ as well as $x_i = \exp(-\lambda_i)$ for $i = 1, \ldots, n$. Then, (4.16) becomes

$$\mathcal{P}(\omega') = x_0 \cdot \prod_{i=1}^{n} x_i^{f_{\Sigma,i}(\omega')}, \qquad \omega' \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma). \tag{4.17}$$

Further, from the normalization condition $\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}(\omega') = 1$ it follows that x_0

has to satisfy

$$x_0 = \left(\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Omega)} \prod_{i=1}^{n} x_i^{f_{\Sigma,i}(\omega)} \right)^{-1},$$

and the dual optimization problem means maximizing $\log(x_0)$ instead of $-1 - \lambda_0$. This can be further simplified as follows. Note that $\log(x_0)$ is maximal if and only if x_0 is maximal, and that x_0 is maximal if and only if x_0^{-1} is minimal. This allows us to transform the dual problem (4.15) once again, namely into minimizing x_0^{-1}, and we obtain the dual maximum entropy optimization problem (4.12). The constraints $x_i > 0$ for $i = 1, \ldots, n$ originate from the substitution $x_i = \exp(-\lambda_i)$. Since the exponential function $\exp(-\lambda_i)$ is positive for all $\lambda_i \in \mathbb{R}$, the x_i's must be positive as well.

Finally, Slater's condition tells us that the solution of the dual optimization problem corresponds to the solution of the primal one. More precisely, if the vector $\vec{\alpha}_{\Sigma,\mathcal{R}} = (\alpha_1, \ldots, \alpha_n)$ solves (4.12), one has

$$\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) = \alpha_0 \cdot \prod_{i=1}^{n} \alpha_i^{f_{\Sigma,i}(\omega)}$$

for $\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$, where $\alpha_0 = (\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \prod_{i=1}^{n} \alpha_i^{f_{\Sigma,i}(\omega)})^{-1}$. □

For most knowledge bases, the Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$ is unique. However, in some rare cases there can be several solutions of the dual maximum entropy optimization problem which all lead to the same optimum, though. This happens when there are redundant conditionals in the knowledge base as illustrated in the next example. Note that this is not a problem because any Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$ is sufficient for (4.13) in this case.

Example 4.3.3

Let $\Sigma_{\mathsf{ex}} = (\emptyset, \{A/0, B/0\})$ be a signature, and let $\mathcal{R}_{\mathsf{ex}} = (\emptyset, \{r_1, r_2\})$ be a knowledge base with $r_1 = (B|A)[p]$ and $r_2 = (AB|A)[p]$ where $p \in (0,1)$. Then, there is one possible world in which both r_1 and r_2 are verified, namely $\omega = AB$, and one possible world in which both r_1 and r_2 are falsified, $\omega = A\overline{B}$. In the remaining possible worlds both conditionals are not applicable. Hence, the dual maximum entropy optimization problem (4.12) becomes

$$\begin{aligned} &\text{minimize} \quad x_1^{1-p} x_2^{1-p} + x_1^{-p} x_2^{-p} \qquad \left(= \frac{x_1 x_2 + 1}{(x_1 x_2)^p} \right) \\ &\text{subject to} \quad x_1, x_2 > 0. \end{aligned}$$

The minimum is obtained whenever $x_1 x_2 = \frac{p}{1-p}$ holds. To see this, consider the function $f(x) = \frac{x+1}{x^p}$ with its derivative $f'(x) = \frac{x^p-(x+1)\cdot p\cdot x^{p-1}}{x^{2p}}$. The derivative $f'(x)$ vanishes in $x = \frac{p}{1-p}$. Hence, the solutions of the optimization problem are all vectors $\vec{\alpha}_{\Sigma_{\mathsf{ex}},\mathcal{R}_{\mathsf{ex}}} = (\alpha_1, \alpha_2)$ with

$$\alpha_1 > 0 \quad \text{and} \quad \alpha_2 = \frac{p}{1-p} \cdot \frac{1}{\alpha_1}.$$

Because in all possible worlds $\omega \in \Omega(\Sigma_{\mathsf{ex}})$, the two conditionals r_1 and r_2 are evaluated the same (both conditionals are either verified, falsified, or not applicable), the factors α_1 and α_2 occur in the maximum entropy probabilities $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}_{\mathsf{ex}}}(\omega)$ always together in form of the product $\alpha_1 \cdot \alpha_2$. This product can be simplified to

$$\alpha_1 \cdot \alpha_2 = \alpha_1 \cdot \frac{p}{1-p} \cdot \frac{1}{\alpha_1} = \frac{p}{1-p}.$$

Hence, it is independent of the single values of α_1 and α_2.

Proposition 4.3.2 tells us that the maximum entropy model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ of a Σ-p-consistent knowledge base $\mathcal{R}$ is fully determined by a Σ-ME-vector $\vec{\alpha}_{\mathcal{R},\Sigma}$ and the set $\Omega^0_{\mathcal{F}_\mathcal{R}}(\Sigma)$ of the possible worlds ω with probability $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) = 0$. In particular, there is no need to compute and store the probabilities $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega)$ for all possible worlds $\omega \in \Omega(\Sigma)$ when drawing inferences from $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$. Instead, one determines and stores the vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$ and calculates the needed probability values on demand. Recall that the size of $\vec{\alpha}_{\Sigma,\mathcal{R}}$ is equal to n, the number of conditionals in $\mathcal{B}_\mathcal{R}$, and therewith constant in $|\mathsf{Const}_\Sigma|$. In general, $|\vec{\alpha}_{\mathcal{R},\Sigma}| \ll |\{\mathcal{P}(\omega) \mid \omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)\}|$ holds and storage costs decrease to a minimum by considering the dual problem.

Besides that, the dual maximum entropy optimization problem yields with (4.13) a product representation of the maximum entropy model. We analyze this product representation in more detail now.

▶ Conditional Indifference of the Maximum Entropy Model

Equation (4.13) proves that the maximum entropy probabilities $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega)$ for $\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$ can be written as a product of the components of the Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$. This product representation is very closely linked to conditional structures. To show this connection, we consider the semiring

$$\mathcal{S}(\mathcal{R}) = \mathcal{S}(\{\delta \mid \delta[p] \in \mathcal{B}_\mathcal{R}\}). \tag{4.18}$$

Recall that

$$\mathcal{S}(\{\delta \mid \delta[p] \in \mathcal{B}_\mathcal{R}\}) = (\mathcal{G}_G(\{\delta \mid \delta[p] \in \mathcal{B}_\mathcal{R}\}), +, \cdot, 1, 0)$$

constitutes the algebraic framework for the conditional structures of the conditionals in $\mathcal{R}$ (cf. (3.4) on page 75). Further, we introduce a homomorphism $\rho_{\vec{p}} : \mathcal{S}(\mathcal{R}) \to \mathbb{R}_{>0}$

which maps elements from $\mathcal{S}(\mathcal{R})$ to positive real numbers while respecting the algebraic structure of $\mathcal{S}(\mathcal{R})$. This homomorphism $\rho_{\vec{p}}$ is parameterized by the vector $\vec{p} = (p_1, \ldots, p_n)$ consisting of the probabilities of the conditionals in $\mathcal{B}_{\mathcal{R}}$ and given by $\rho_{\vec{p}}(\mathsf{a}_i^+) = \alpha_i^{1-p_i}$ and $\rho_{\vec{p}}(\mathsf{a}_i^-) = \alpha_i^{-p_i}$ for the generators a_i^+ and a_i^-, $i = 1, \ldots, n$, of $\mathcal{S}(\mathcal{R})$ as well as

$$\rho_{\vec{p}}(\sum_{h=1}^{l}\prod_{i=1}^{n}(\mathsf{a}_i^+)^{m_i(h)} \cdot (\mathsf{a}_i^-)^{k_i(h)}) = \sum_{h=1}^{l}\prod_{i=1}^{n}\rho_{\vec{p}}(\mathsf{a}_i^+)^{m_i(h)} \cdot \rho_{\vec{p}}(\mathsf{a}_i^-)^{k_i(h)}$$

for compounded elements from $\mathcal{S}(\mathcal{R})$. By applying this homomorphism $\rho_{\vec{p}}$ we can abstract from the specific probability values in $\vec{p}$ and highlight on the influence of the conditional structures on the probabilities $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega)$. We obtain the following proposition.

Proposition 4.3.4: Maximum Entropy and Conditional Structures
Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a Σ-p-consistent knowledge base, and let $\vec{p} = (p_1, \ldots, p_n)$ be the vector of the probabilities of the conditionals in $\mathcal{B}_{\mathcal{R}}$. Then,

$$\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) = \begin{cases} \dfrac{\rho_{\vec{p}}(\sigma_{\Sigma,\mathcal{B}_{\mathcal{R}}}(\omega))}{\rho_{\vec{p}}(\sigma_{\Sigma,\mathcal{B}_{\mathcal{R}}}(\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)))}, & \text{if} \quad \omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma) \\ 0 & \text{otherwise} \end{cases}.$$

Proof. For $\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$, we have

$$\begin{aligned}
&\rho_{\vec{p}}(\sigma_{\Sigma,\mathcal{B}_{\mathcal{R}}}(\omega)) \\
&= \rho_{\vec{p}}(\prod_{i=1}^{n}(\mathsf{a}_i^+)^{\mathsf{ver}_{\Sigma,r_i}(\omega)} \cdot (\mathsf{a}_i^-)^{\mathsf{fal}_{\Sigma,r_i}(\omega)}) && = \prod_{i=1}^{n}\rho_{\vec{p}}((\mathsf{a}_i^+)^{\mathsf{ver}_{\Sigma,r_i}(\omega)}) \cdot \rho_{\vec{p}}((\mathsf{a}_i^-)^{\mathsf{fal}_{\Sigma,r_i}(\omega)}) \\
&= \prod_{i=1}^{n}\rho_{\vec{p}}(\mathsf{a}_i^+)^{\mathsf{ver}_{\Sigma,r_i}(\omega)} \cdot \rho_{\vec{p}}(\mathsf{a}_i^-)^{\mathsf{fal}_{\Sigma,r_i}(\omega)} && = \prod_{i=1}^{n}(\alpha_i^{1-p_i})^{\mathsf{ver}_{\Sigma,r_i}(\omega)} \cdot (\alpha_i^{-p_i})^{\mathsf{fal}_{\Sigma,r_i}(\omega)} \\
&= \prod_{i=1}^{n}\alpha_i^{\mathsf{ver}_{\Sigma,r_i}(\omega) - p_i \cdot \mathsf{app}_{\Sigma,r_i}(\omega)} && = \prod_{i=1}^{n}\alpha_i^{f_{\Sigma,r_i}(\omega)},
\end{aligned}$$

and, consequently,

$$\begin{aligned}
&\rho_{\vec{p}}(\sigma_{\Sigma,\mathcal{B}_{\mathcal{R}}}(\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma))) \\
&= \rho_{\vec{p}}(\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)}\sigma_{\Sigma,\mathcal{B}_{\mathcal{R}}}(\omega)) = \sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)}\rho_{\vec{p}}(\sigma_{\Sigma,\mathcal{B}_{\mathcal{R}}}(\omega)) = \sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)}\prod_{i=1}^{n}\alpha_i^{f_{\Sigma,r_i}(\omega)}.
\end{aligned}$$

Therewith,

$$\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) = \begin{cases} \frac{\prod_{i=1}^{n} \alpha_i^{f_{\Sigma,r_i}(\omega)}}{\sum_{\omega\in\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} \prod_{i=1}^{n} \alpha_i^{f_{\Sigma,r_i}(\omega)}} & \text{if} \quad \omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma) \\ 0 & \text{otherwise} \end{cases}$$
$$= \begin{cases} \frac{\rho_{\vec{p}}(\sigma_{\Sigma,\mathcal{B}_\mathcal{R}}(\omega))}{\rho_{\vec{p}}(\sigma_{\Sigma,\mathcal{B}_\mathcal{R}}(\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)))}, & \text{if} \quad \omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma) \\ 0 & \text{otherwise} \end{cases}$$

follows. □

Proposition 4.3.4 shows that computing maximum entropy probabilities compares with computing conditional structures. As a consequence, if one is able to find a compact representation of conditional structures, then this leads to a compact representation of the maximum entropy probabilities as well (cf. Section 6.4). This justifies our investigations on efficient computations of conditional structures in Section 3.2 and Section 6.4.

From the product representation of the maximum entropy probabilities in (4.13), it directly follows that conditionally equivalent (positive) possible worlds have the same maximum entropy probability.

Proposition 4.3.5: Conditional Indifference
Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_\mathcal{R}, \mathcal{B}_\mathcal{R})$ be a Σ-p-consistent knowledge base, and let $\omega, \omega' \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$ be possible worlds. Then,

$$\omega \sim_{\mathcal{B}_\mathcal{R}} \omega' \quad \text{implies} \quad \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) = \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega').$$

Proof. According to (4.13) and Proposition 3.2.7, we have

$$\begin{aligned} &\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) \\ &= \alpha_0 \cdot \prod_{i=1}^{n} \alpha_i^{f_{\Sigma,i}(\omega)} && = \alpha_0 \cdot \prod_{i=1}^{n} \alpha_i^{\mathsf{ver}_{\Sigma,i}(\omega) - p\cdot\mathsf{app}_{\Sigma,i}(\omega)} \\ &= \alpha_0 \cdot \prod_{i=1}^{n} \alpha_i^{\mathsf{ver}_{\Sigma,i}(\omega) - p\cdot(\mathsf{ver}_{\Sigma,i}(\omega)+\mathsf{fal}_{\Sigma,i}(\omega))} && = \alpha_0 \cdot \prod_{i=1}^{n} \alpha_i^{\mathsf{ver}_{\Sigma,i}(\omega') - p\cdot(\mathsf{ver}_{\Sigma,i}(\omega')+\mathsf{fal}_{\Sigma,i}(\omega'))} \\ &= \alpha_0 \cdot \prod_{i=1}^{n} \alpha_i^{\mathsf{ver}_{\Sigma,i}(\omega') - p\cdot\mathsf{app}_{\Sigma,i}(\omega')} && = \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega') \end{aligned}$$

for $\omega, \omega' \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$ with $\omega \sim_{\mathcal{B}_\mathcal{R}} \omega'$. □

Note that the implication in Proposition 4.3.5 does not hold the other way around. That is, the MaxEnt probabilities of two possible worlds $\omega, \omega' \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$

might be the same although ω and ω' differ in their conditional structures as the next example shows.

Example 4.3.6

A very simple example which shows that possible worlds with different conditional structures may have the same maximum entropy probability is provided by the knowledge base $\mathcal{R}_{\mathsf{ex}} = (\emptyset, \{(A|\top)[0.5]\})$ which is defined over the signature $\Sigma_{\mathsf{ex}} = (\emptyset, \{A/0\})$. Then, for the conditional structures of the only two possible worlds A and $\overline{A}$ it holds that $\sigma_{\Sigma_{\mathsf{ex}},\mathcal{R}_{\mathsf{ex}}}(A) = \mathsf{a}_1^+ \neq \mathsf{a}_1^- = \sigma_{\Sigma_{\mathsf{ex}},\mathcal{R}_{\mathsf{ex}}}(\overline{A})$. However, the maximum entropy probabilities of these possible worlds agree as it is $\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{ex}},\mathcal{R}_{\mathsf{ex}}}(A) = 0.5 = \mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{ex}},\mathcal{R}_{\mathsf{ex}}}(\overline{A})$. Note that this is the only model of $\mathcal{R}_{\mathsf{ex}}$, thus, its maximum entropy model.

In general, there is no closed-form expression for the Σ-ME-vector(s) $\vec{\alpha}_{\mathcal{R},\Sigma}$. In some rare cases, however, it is possible to derive $\vec{\alpha}_{\mathcal{R},\Sigma}$ analytically (cf. [Kern-Isberner, 2001a]) by solving a non-linear equation system. The next paragraph deals with this equation system.

▶ Maximum Entropy Equation System

In this paragraph, we consider an equation system-based method for computing the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ of a Σ-p-consistent knowledge base $\mathcal{R}$. We call the respective equation system *maximum entropy equation system*. To be more precise, a solution of this equation system is not the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ directly but a Σ-ME-vector $\vec{\alpha}_{\mathcal{R},\Sigma}$ which determines $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ according to (4.13). Besides the primal and the dual maximum entropy optimization problem, the maximum entropy equation system provides a third way of computing $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$.

Technically, the maximum entropy equation system can be derived from the characterization of the aggregating semantics (3.11) when recognizing that the condition

$$\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathsf{app}_{\Sigma,r}(\omega) \cdot \mathcal{P}(\omega) > 0, \qquad r \in \mathcal{B}_{\mathcal{R}},$$

is trivially satisfied by the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ if the knowledge base $\mathcal{R}$ is Σ-p-consistent (cf. Proposition 4.2.11).

Definition 4.3.7: Maximum Entropy Equation System

Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a Σ-p-consistent knowledge base, and let $f_{\Sigma,i}$, $i = 1, \ldots, n$, be the feature functions with respect to the conditionals $r_i \in \mathcal{B}_{\mathcal{R}}$. Then, we call the equation system

$$\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} f_{\Sigma,j}(\omega) \cdot \prod_{i=1}^{n} x_i^{f_{\Sigma,i}(\omega)} = 0, \qquad j = 1, \ldots, n. \tag{4.19}$$

the *maximum entropy equation system* of $\mathcal{R}$ with respect to Σ.

The maximum entropy equation system (4.19) is the first condition of the characterization of the aggregating semantics in (3.11) applied to the conditionals in $\mathcal{B}_{\mathcal{R}}$ in which the product representation of the maximum entropy probabilities (4.13) is plugged in and in which $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) = 0$ for $\omega \in \Omega^0_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ is considered such that the sums range over the possible worlds in $\Omega_{\mathcal{F}_{\mathcal{R}}(\Sigma)}$ only. If a vector $\vec{\beta} \in \mathbb{R}^n_{>0}$ is a solution of (4.19), then $\vec{\beta}$ determines the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ as the next proposition shows. As a consequence, if a probability distribution is a model of $\mathcal{R}$ and has a product representation of the form (4.13), then it is automatically $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$.

Proposition 4.3.8: Correctness of Maximum Entropy Equation System

Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a Σ-p-consistent knowledge base, let $f_{\Sigma,i}$, $i = 1, \ldots, n$, be the feature functions with respect to $r_i \in \mathcal{B}_{\mathcal{R}}$, and let $\vec{\beta} = (\beta_1, \ldots, \beta_n) \in \mathbb{R}^n_{>0}$ be a vector of positive real numbers. Then, the function

$$\mathcal{P}_{\Sigma,\vec{\beta}}(\omega) = \begin{cases} \beta_0 \cdot \prod_{i=1}^{n} \beta_i^{f_{\Sigma,i}(\omega)} & \text{if} \quad \omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma) \\ 0 & \text{if} \quad \omega \in \Omega^0_{\mathcal{F}_{\mathcal{R}}}(\Sigma) \end{cases},$$

where

$$\beta_0 = \Big(\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \prod_{i=1}^{n} \beta_i^{f_{\Sigma,i}(\omega)} \Big)^{-1}$$

is a normalizing constant, is a probability distribution over $\Omega(\Sigma)$. If $\vec{\beta}$ is a solution of the maximum entropy equation system (4.19) in addition, then $\mathcal{P}_{\Sigma,\vec{\beta}}(\omega) = \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega)$ for $\omega \in \Omega(\Sigma)$ holds. This also applies when the feature functions $f_{\Sigma,i}$ are replaced by the standardized feature functions $\hat{f}_{\Sigma,i}$ (cf. Definition 3.4.10).

Proof. The fact that $\mathcal{P}_{\Sigma,\vec{\beta}}$ is a probability distribution is trivial. Since $\beta_i > 0$ for $i = 1, \ldots, n$, it holds that $\prod_{i=1}^{n} \beta_i^{f_{\Sigma,i}(\omega)} > 0$ as well. Due to the normalization with β_0, both $\mathcal{P}_{\Sigma,\vec{\beta}}(\omega) \in [0,1]$ for $\omega \in \Omega(\Sigma)$ and $\sum_{\omega \in \Omega(\Sigma)} \mathcal{P}_{\Sigma,\vec{\beta}}(\omega) = 1$ immediately follow. Also by definition, $\mathcal{P}_{\Sigma,\vec{\beta}}$ is a Σ-model of $\mathcal{R}$ if $\vec{\beta}$ is a solution of (4.19) because it follows that all conditionals in $\mathcal{B}_{\mathcal{R}}$ are accepted according to (3.11) then. Further, the condition $\mathcal{P}_{\Sigma,\vec{\beta}}(\omega) = 0$ for $\omega \in \Omega^0_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ guarantees that $\mathcal{P}_{\Sigma,\vec{\beta}}$ satisfies the facts in $\mathcal{F}_{\mathcal{R}}$, too.

The rest of the proof is an elaboration of ideas from [Darroch and Ratcliff, 1972]. One has

$$\begin{aligned}
\mathcal{H}(\mathcal{P}_{\Sigma,\vec{\beta}}) &= -\sum_{\omega\in\Omega(\Sigma)} \mathcal{P}_{\Sigma,\vec{\beta}}(\omega)\cdot\log(\mathcal{P}_{\Sigma,\vec{\beta}}(\omega)) \\
&= -\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}_{\Sigma,\vec{\beta}}(\omega)\cdot\log(\mathcal{P}_{\Sigma,\vec{\beta}}(\omega)) \\
&= -\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}_{\Sigma,\vec{\beta}}(\omega)\cdot\log(\beta_0\cdot\prod_{i=1}^{n}\beta_i^{f_{\Sigma,i}(\omega)}) \\
&= -\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}_{\Sigma,\vec{\beta}}(\omega)\cdot(\log(\beta_0)+\sum_{i=1}^{n} f_{\Sigma,i}(\omega)\cdot\log(\beta_i)) \\
&= -\log(\beta_0)\cdot\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}_{\Sigma,\vec{\beta}}(\omega) - \sum_{i=1}^{n}\log(\beta_i)\cdot\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} f_{\Sigma,i}(\omega)\cdot\mathcal{P}_{\Sigma,\vec{\beta}}(\omega) \\
&\overset{(*)}{=} -\log(\beta_0)\cdot\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) - \sum_{i=1}^{n}\log(\beta_i)\cdot\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} f_{\Sigma,i}(\omega)\cdot\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) \\
&= -\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega)\cdot\log(\beta_0\cdot\prod_{i=1}^{n}\beta_i^{f_{\Sigma,i}(\omega)}) \\
&= -\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega)\cdot\log(\mathcal{P}_{\Sigma,\vec{\beta}}(\omega)) \\
&= -\sum_{\omega\in\Omega(\Sigma)} \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega)\cdot\log(\mathcal{P}_{\Sigma,\vec{\beta}}(\omega)).
\end{aligned}$$

The equality $(*)$ holds because of the normalization condition of probability distributions which leads to

$$\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}_{\Sigma,\vec{\beta}}(\omega) = 1 = \sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega),$$

and because $\vec{\beta}$ satisfies (4.19) and, thus,

$$\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} f_{\Sigma,i}(\omega)\cdot\mathcal{P}_{\Sigma,\vec{\beta}}(\omega) = 0 = \sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} f_{\Sigma,i}(\omega)\cdot\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega).$$

Consequently, the difference of the entropies of $\mathcal{P}_{\Sigma,\vec{\beta}}$ and $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ yields

$$\begin{aligned}
&\mathcal{H}(\mathcal{P}_{\Sigma,\vec{\beta}}) - \mathcal{H}(\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}) \\
&= -\sum_{\omega\in\Omega(\Sigma)} \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega)\cdot\log(\mathcal{P}_{\Sigma,\vec{\beta}}(\omega)) + \sum_{\omega\in\Omega(\Sigma)} \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega)\cdot\log(\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega))
\end{aligned}$$

ω	$\sigma_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(\omega)$	$\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(\omega)$	ω	$\sigma_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(\omega)$	$\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}}(\omega)$
BSR	a_1^+	0.229	$\overline{B}SR$	a_1^-	0.153
$BS\overline{R}$	a_2^-	0.002	$\overline{B}S\overline{R}$	a_2^+	0.204
$B\overline{S}R$	a_2^-	0.002	$\overline{B}\overline{S}R$	a_2^+	0.204
$B\overline{S}\overline{R}$	a_2^-	0.002	$\overline{B}\overline{S}\overline{R}$	a_2^+	0.204

Table 4.1: Conditional structures and maximum entropy probabilities rounded to three decimal places of the possible worlds in $\Omega(\Sigma_{\mathsf{bow}})$ with respect to $\mathcal{R}_{\mathsf{bow}}$ from Example 4.3.9.

$$
\begin{aligned}
&= \sum_{\omega\in\Omega(\Sigma)} \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega)\cdot\left(\log(\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega)) - \log(\mathcal{P}_{\Sigma,\vec{\beta}}(\omega))\right)\\
&= \sum_{\omega\in\Omega(\Sigma)} \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega)\cdot\log\left(\frac{\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega)}{\mathcal{P}_{\Sigma,\vec{\beta}}(\omega)}\right)\\
&= \mathcal{KL}(\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}},\mathcal{P}_{\Sigma,\vec{\beta}})
\end{aligned}
$$

where $\mathcal{KL}(\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}},\mathcal{P}_{\Sigma,\vec{\beta}})$ is the *Kullback-Leibler divergence* (cf. [Kullback and Leibler, 1951; Kullback, 1959] and Definition A.1.7 in the appendix for a definition of the Kullback-Leibler divergence). Because the Kullback-Leibler divergence is known to be non-negative, one has

$$\mathcal{H}(\mathcal{P}_{\Sigma,\vec{\beta}}) - \mathcal{H}(\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}) \geq 0.$$

Thus, $\mathcal{P}_{\Sigma,\vec{\beta}}$ has an entropy greater than or equal to $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ from which $\mathcal{H}(\mathcal{P}_{\Sigma,\vec{\beta}}) = \mathcal{H}(\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}})$ follows, because $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ maximizes the entropy. Further, because $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ is the unique probabilistic Σ-model of $\mathcal{R}$ with maximal entropy, from $\mathcal{H}(\mathcal{P}_{\Sigma,\vec{\beta}}) = \mathcal{H}(\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}})$ it follows that $\mathcal{P}_{\Sigma,\vec{\beta}} = \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ must hold. ☐

In general, the maximum entropy equation system (4.19) can be solved numerically only. However, there are some cases in which it can be solved analytically as well. In the following, we give examples for that. The first example is rather simple and solving the equation system is possible because it decomposes into two independent equations which can be solved independently.

Example 4.3.9

We consider the signature $\Sigma_{\mathsf{bow}} = (\emptyset, \{B/0, S/0, R/0\})$ and the knowledge base

$$\mathcal{R}_{\mathsf{bow}} = (\emptyset, \{(B|SR)[0.6],\ (\overline{B}|\overline{S}\vee\overline{R})[0.99]\})$$

from Example 4.1.3. The conditional structures of the possible worlds

ω	$\sigma_{\Sigma_{\mathsf{ex}}^m, r_1^m}(\omega)$	$\sigma_{\Sigma_{\mathsf{ex}}^m, r_2^m}(\omega)$
$A(c_m)B(c_m)C(c_m)$	a_1^+	a_2^+
$A(c_m)B(c_m)\overline{C(c_m)}$	a_1^+	a_2^-
$A(c_m)\overline{B(c_m)}C(c_m)$	a_1^-	1
$A(c_m)\overline{B(c_m)}\overline{C(c_m)}$	a_1^-	1
$\overline{A(c_m)}B(c_m)C(c_m)$	1	a_2^+
$\overline{A(c_m)}B(c_m)\overline{C(c_m)}$	1	a_2^-
$\overline{A(c_m)}\overline{B(c_m)}C(c_m)$	1	1
$\overline{A(c_m)}\overline{B(c_m)}\overline{C(c_m)}$	1	1

Table 4.2: Conditional structures of the possible worlds $\omega \in \Omega(\Sigma_{\mathsf{ex}}^m)$ with respect to $\Sigma_{\mathsf{ex}}^m = (\{c_m\}, \{A/1, B/1, C/1\})$ and the conditionals $r_1^m = (B(c_m)|A(c_m))[p_1]$ and $r_2^m = (C(c_m)|B(c_m))[p_2]$ from Example 4.3.10).

in $\Omega(\Sigma_{\mathsf{bow}})$ with respect to $\mathcal{R}_{\mathsf{bow}}$ are shown in Table 4.1. For this particular example, the maximum entropy equation system (4.19) is

$$\begin{aligned} (1-0.6)\cdot\alpha_1^{1-0.6} - 0.6\cdot\alpha_1^{-0.6} &= 0 \\ 3\cdot(1-0.99)\cdot\alpha_2^{1-0.99} - 3\cdot 0.99\cdot\alpha_2^{-0.99} &= 0. \end{aligned}$$

Multiplying both sides with $\alpha_1^{0.6}$ and $\alpha_2^{0.99}$, respectively, and rearranging the equations results in

$$\alpha_1 = \frac{0.6}{1-0.6} = \frac{3}{2}, \quad \text{and} \quad \alpha_2 = \frac{3\cdot 0.99}{3\cdot(1-0.99)} = 99,$$

and, hence, in the Σ_{bow}-ME-vector $\vec{\alpha}_{\Sigma_{\mathsf{bow}},\mathcal{R}_{\mathsf{bow}}} = (1.5, 99)$. The resulting maximum entropy model of $\mathcal{R}_{\mathsf{bow}}$ is also shown in Table 4.1.

The second example is more advanced and considers predicates of arity greater than zero as well as constants. In order to find an analytical solution of the maximum entropy equation system (4.19), we apply sophisticated manipulations on the equations which capture the idea of splitting and rearranging sums of conditional structures as suggested in Example 3.2.4.

Example 4.3.10

We consider the knowledge base $\mathcal{R}_{\mathsf{ex}} = (\emptyset, \{r_1, r_2\})$ with

$$r_1 = (B(X)|A(X))[p_1], \qquad r_2 = (C(X)|B(X))[p_2],$$

and $p_1, p_2 \in (0,1)$ which is defined over the signature $\Sigma_{\mathsf{ex}} = (\mathsf{Const}_{\mathsf{ex}}, \mathsf{Pred}_{\mathsf{ex}})$ given by $\mathsf{Const}_{\Sigma_{\mathsf{ex}}} = \{c_1, c_2, c_3\}$ and $\mathsf{Pred}_{\mathsf{ex}} = \{A/1, B/1, C/1\}$. Analogously to Example 3.2.4, we define the subsignatures $\Sigma_{\mathsf{ex}}^l = (\{c_l\}, \mathsf{Pred}_{\mathsf{ex}})$ for $l = 1, 2, 3$, and refer to the instances of the conditionals r_1 and r_2 following the schema $r_1^l = (B(c_l)|A(c_l))[p_1]$ and $r_2^l = (B(c_l)|A(c_l))[p_2]$. According to (3.11), for $j = 1, 2$, we have $\mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{ex}}, \mathcal{R}_{\mathsf{ex}}} \models r_j$ if and only if

$$\begin{aligned}
0 &= \sum_{\omega \in \Omega(\Sigma_{\mathsf{ex}})} f_{\Sigma_{\mathsf{ex}}, r_j}(\omega) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma_{\mathsf{ex}}, r_i}(\omega)} \\
&= \sum_{\omega_1 \in \Omega(\Sigma^1_{\mathsf{ex}})} \sum_{\omega_2 \in \Omega(\Sigma^2_{\mathsf{ex}})} \sum_{\omega_3 \in \Omega(\Sigma^3_{\mathsf{ex}})} f_{\Sigma_{\mathsf{ex}}, r_j}(\omega_1\omega_2\omega_3) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma_{\mathsf{ex}}, r_i}(\omega_1\omega_2\omega_3)}.
\end{aligned}$$

With Proposition 3.4.4, it follows that

$$0 = \sum_{\omega_1 \in \Omega(\Sigma^1_{\mathsf{ex}})} \sum_{\omega_2 \in \Omega(\Sigma^2_{\mathsf{ex}})} \sum_{\omega_3 \in \Omega(\Sigma^3_{\mathsf{ex}})} \left(\sum_{l=1}^{3} f_{\Sigma_{\mathsf{ex}}, r_j^l}(\omega_1\omega_2\omega_3) \right) \cdot \prod_{i=1,2} \alpha_i^{\sum_{l=1}^3 f_{\Sigma_{\mathsf{ex}}, r_i^l}(\omega_1\omega_2\omega_3)}.$$

Now, for $l \in \{1, 2, 3\}$, because the instance r_j^l of r_j uses atoms from Σ^l_{ex} only, we have $f_{\Sigma_{\mathsf{ex}}, r_j^l}(\omega_1\omega_2\omega_3) = f_{\Sigma^l_{\mathsf{ex}}, r_j^l}(\omega_l)$ and, hence,

$$0 = \sum_{\omega_1 \in \Omega(\Sigma^1_{\mathsf{ex}})} \sum_{\omega_2 \in \Omega(\Sigma^2_{\mathsf{ex}})} \sum_{\omega_3 \in \Omega(\Sigma^3_{\mathsf{ex}})} \left(\sum_{l=1}^{3} f_{\Sigma^l_{\mathsf{ex}}, r_j^l}(\omega_l) \right) \cdot \prod_{i=1,2} \alpha_i^{\sum_{l=1}^3 f_{\Sigma^l_{\mathsf{ex}}, r_i^l}(\omega_l)}.$$

As a consequence, we may rearrange the sums and factor out so that

$$\begin{aligned}
0 = \sum_{l=1}^{3} & \sum_{\omega_1 \in \Omega(\Sigma^1_{\mathsf{ex}})} \sum_{\omega_2 \in \Omega(\Sigma^2_{\mathsf{ex}})} \sum_{\omega_3 \in \Omega(\Sigma^3_{\mathsf{ex}})} f_{\Sigma^l_{\mathsf{ex}}, r_j^l}(\omega_l) \cdot \prod_{i=1,2} \prod_{m=1}^{3} \alpha_i^{f_{\Sigma^m_{\mathsf{ex}}, r_i^m}(\omega_m)} \\
= & \sum_{\omega_1 \in \Omega(\Sigma^1_{\mathsf{ex}})} \sum_{\omega_2 \in \Omega(\Sigma^2_{\mathsf{ex}})} \sum_{\omega_3 \in \Omega(\Sigma^3_{\mathsf{ex}})} f_{\Sigma^1_{\mathsf{ex}}, r_j^1}(\omega_1) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma^1_{\mathsf{ex}}, r_i^1}(\omega_1)} \alpha_i^{f_{\Sigma^2_{\mathsf{ex}}, r_i^2}(\omega_2)} \alpha_i^{f_{\Sigma^3_{\mathsf{ex}}, r_i^3}(\omega_3)} \\
& + \sum_{\omega_1 \in \Omega(\Sigma^1_{\mathsf{ex}})} \sum_{\omega_2 \in \Omega(\Sigma^2_{\mathsf{ex}})} \sum_{\omega_3 \in \Omega(\Sigma^3_{\mathsf{ex}})} f_{\Sigma^2_{\mathsf{ex}}, r_j^2}(\omega_2) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma^1_{\mathsf{ex}}, r_i^1}(\omega_1)} \alpha_i^{f_{\Sigma^2_{\mathsf{ex}}, r_i^2}(\omega_2)} \alpha_i^{f_{\Sigma^3_{\mathsf{ex}}, r_i^3}(\omega_3)} \\
& + \sum_{\omega_1 \in \Omega(\Sigma^1_{\mathsf{ex}})} \sum_{\omega_2 \in \Omega(\Sigma^2_{\mathsf{ex}})} \sum_{\omega_3 \in \Omega(\Sigma^3_{\mathsf{ex}})} f_{\Sigma^3_{\mathsf{ex}}, r_j^3}(\omega_3) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma^1_{\mathsf{ex}}, r_i^1}(\omega_1)} \alpha_i^{f_{\Sigma^2_{\mathsf{ex}}, r_i^2}(\omega_2)} \alpha_i^{f_{\Sigma^3_{\mathsf{ex}}, r_i^3}(\omega_3)}
\end{aligned}$$

$$
\begin{aligned}
= & \sum_{\omega_1 \in \Omega(\Sigma^1_{\mathsf{ex}})} f_{\Sigma^1_{\mathsf{ex}}, r^1_j}(\omega_1) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma^1_{\mathsf{ex}}, r^1_i}(\omega_1)} \\
& \cdot \left(\sum_{\omega_2 \in \Omega(\Sigma^2_{\mathsf{ex}})} \prod_{i=1,2} \alpha_i^{f_{\Sigma^2_{\mathsf{ex}}, r^2_i}(\omega_2)} \cdot \left(\sum_{\omega_3 \in \Omega(\Sigma^3_{\mathsf{ex}})} \prod_{i=1,2} \alpha_i^{f_{\Sigma^3_{\mathsf{ex}}, r^3_i}(\omega_3)} \right) \right) \\
& + \sum_{\omega_1 \in \Omega(\Sigma^1_{\mathsf{ex}})} \prod_{i=1,2} \alpha_i^{f_{\Sigma^1_{\mathsf{ex}}, r^1_i}(\omega_1)} \\
& \cdot \left(\sum_{\omega_2 \in \Omega(\Sigma^2_{\mathsf{ex}})} f_{\Sigma^2_{\mathsf{ex}}, r^2_j}(\omega_2) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma^2_{\mathsf{ex}}, r^2_i}(\omega_2)} \cdot \left(\sum_{\omega_3 \in \Omega(\Sigma^3_{\mathsf{ex}})} \prod_{i=1,2} \alpha_i^{f_{\Sigma^3_{\mathsf{ex}}, r^3_i}(\omega_3)} \right) \right) \\
& + \sum_{\omega_1 \in \Omega(\Sigma^1_{\mathsf{ex}})} \prod_{i=1,2} \alpha_i^{f_{\Sigma^1_{\mathsf{ex}}, r^1_i}(\omega_1)} \\
& \cdot \left(\sum_{\omega_2 \in \Omega(\Sigma^2_{\mathsf{ex}})} \prod_{i=1,2} \alpha_i^{f_{\Sigma^2_{\mathsf{ex}}, r^2_i}(\omega_2)} \cdot \left(\sum_{\omega_3 \in \Omega(\Sigma^3_{\mathsf{ex}})} f_{\Sigma^3_{\mathsf{ex}}, r^3_j}(\omega_3) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma^3_{\mathsf{ex}}, r^3_i}(\omega_3)} \right) \right) \\
= & \left(\sum_{\omega_1 \in \Omega(\Sigma^1_{\mathsf{ex}})} f_{\Sigma^1_{\mathsf{ex}}, r^1_j}(\omega_1) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma^1_{\mathsf{ex}}, r^1_i}(\omega_1)} \right) \cdot \left(\sum_{\omega_2 \in \Omega(\Sigma^2_{\mathsf{ex}})} \prod_{i=1,2} \alpha_i^{f_{\Sigma^2_{\mathsf{ex}}, r^2_i}(\omega_2)} \right) \\
& \cdot \left(\sum_{\omega_3 \in \Omega(\Sigma^3_{\mathsf{ex}})} \prod_{i=1,2} \alpha_i^{f_{\Sigma^3_{\mathsf{ex}}, r^3_i}(\omega_3)} \right) \\
& + \left(\sum_{\omega_1 \in \Omega(\Sigma^1_{\mathsf{ex}})} \prod_{i=1,2} \alpha_i^{f_{\Sigma^1_{\mathsf{ex}}, r^1_i}(\omega_1)} \right) \cdot \left(\sum_{\omega_2 \in \Omega(\Sigma^2_{\mathsf{ex}})} f_{\Sigma^2_{\mathsf{ex}}, r^2_j}(\omega_2) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma^2_{\mathsf{ex}}, r^2_i}(\omega_2)} \right) \\
& \cdot \left(\sum_{\omega_3 \in \Omega(\Sigma^3_{\mathsf{ex}})} \prod_{i=1,2} \alpha_i^{f_{\Sigma^3_{\mathsf{ex}}, r^3_i}(\omega_3)} \right) \\
& + \left(\sum_{\omega_1 \in \Omega(\Sigma^1_{\mathsf{ex}})} \prod_{i=1,2} \alpha_i^{f_{\Sigma^1_{\mathsf{ex}}, r^1_i}(\omega_1)} \right) \cdot \left(\sum_{\omega_2 \in \Omega(\Sigma^2_{\mathsf{ex}})} \prod_{i=1,2} \alpha_i^{f_{\Sigma^2_{\mathsf{ex}}, r^2_i}(\omega_2)} \right) \\
& \cdot \left(\sum_{\omega_3 \in \Omega(\Sigma^3_{\mathsf{ex}})} f_{\Sigma^3_{\mathsf{ex}}, r^3_j}(\omega_3) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma^3_{\mathsf{ex}}, r^3_i}(\omega_3)} \right) \\
= & \sum_{l=1}^{3} \left(\prod_{\substack{m=1,\ldots,3: \\ m \neq l}} \sum_{\omega_m \in \Omega(\Sigma^m_{\mathsf{ex}})} \prod_{i=1,2} \alpha_i^{f_{\Sigma^m_{\mathsf{ex}}, r^m_i}(\omega_m)} \right) \\
& \cdot \left(\sum_{\omega_l \in \Omega(\Sigma^l_{\mathsf{ex}})} f_{\Sigma^l_{\mathsf{ex}}, r^l_j}(\omega_l) \cdot \prod_{i=1,2} \alpha_i^{\Sigma^l_{\mathsf{ex}}, f_{r^l_i}(\omega_l)} \right).
\end{aligned}
$$

Note that the instances of r_1 and r_2 are isomorphic, i.e., for both $j = 1$ and $j = 2$ it holds that, for $l, m \in \{1, 2, 3\}$, the conditionals r_j^l and r_j^m are the same up to replacing the constant c_l in r_j^l by c_m. The same holds for the possible worlds in $\Omega(\Sigma_{\mathsf{ex}}^l)$ and $\Omega(\Sigma_{\mathsf{ex}}^m)$ which leads to the fact that the conditional structures of the possible worlds in $\Omega(\Sigma_{\mathsf{ex}}^l)$ with respect to r_j^l are the same as the conditional structures of the possible worlds in $\Omega(\Sigma_{\mathsf{ex}}^m)$ with respect to r_j^m, i.e.,

$$\sigma_{\Sigma_{\mathsf{ex}}^m,\{r_1^m,r_2^m\}}(\Omega(\Sigma_{\mathsf{ex}}^m)) = \sigma_{\Sigma_{\mathsf{ex}}^l,\{r_1^l,r_2^l\}}(\Omega(\Sigma_{\mathsf{ex}}^l)), \qquad m, l \in \{1, 2, 3\}.$$

This again carries over to the feature functions $f_{\Sigma_{\mathsf{ex}}^l,r_j^l}$ and $f_{\Sigma_{\mathsf{ex}}^m,r_j^m}$ which means that

$$\sum_{\omega_l \in \Omega(\Sigma_{\mathsf{ex}}^l)} \prod_{i=1,2} \alpha_i^{f_{\Sigma_{\mathsf{ex}}^l,r_i^l}(\omega_l)} = \sum_{\omega_m \in \Omega(\Sigma_{\mathsf{ex}}^m)} \prod_{i=1,2} \alpha_i^{f_{\Sigma_{\mathsf{ex}}^m,r_i^m}(\omega_m)}$$

as well as

$$\sum_{\omega_l \in \Omega(\Sigma_{\mathsf{ex}}^l)} f_{\Sigma_{\mathsf{ex}}^l,r_j^l}(\omega_l) \cdot \prod_{i=1,2} \alpha_i^{\Sigma_{\mathsf{ex}}^l,f_{r_i^l}(\omega_l)} = \sum_{\omega_m \in \Omega(\Sigma_{\mathsf{ex}}^m)} f_{\Sigma_{\mathsf{ex}}^m,r_j^m}(\omega_m) \cdot \prod_{i=1,2} \alpha_i^{\Sigma_{\mathsf{ex}}^m,f_{r_i^m}(\omega_m)}$$

holds for $l, m \in \{1, \ldots, k\}$ and $j = 1, 2$. Consequently, we have

$$\begin{aligned} 0 = {} & \\ & \sum_{l=1}^{3} \left(\prod_{\substack{m=1,\ldots,k:\\ m \neq l}} \sum_{\omega_m \in \Omega(\Sigma_{\mathsf{ex}}^m)} \prod_{i=1,2} \alpha_i^{f_{\Sigma_{\mathsf{ex}}^m,r_i^m}(\omega_m)} \right) \cdot \left(\sum_{\omega_l \in \Omega(\Sigma_{\mathsf{ex}}^l)} f_{\Sigma_{\mathsf{ex}}^l,r_j^l}(\omega_l) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma_{\mathsf{ex}}^l,r_i^l}(\omega_l)} \right) \\ & = \underbrace{3 \cdot \left(\sum_{\omega_m \in \Omega(\Sigma_{\mathsf{ex}}^m)} \prod_{i=1,2} \alpha_i^{f_{\Sigma_{\mathsf{ex}}^m,r_i^m}(\omega_m)} \right)^{3-1}}_{>0} \cdot \left(\sum_{\omega_m \in \Omega(\Sigma_{\mathsf{ex}}^m)} f_{\Sigma_{\mathsf{ex}}^m,r_j^m}(\omega_m) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma_{\mathsf{ex}}^m,r_i^m}(\omega_m)} \right). \end{aligned}$$

Because $0 = a \cdot b$ for $a, b \in \mathbb{R}$ with $a > 0$ only holds if $b = 0$, it must be the case that

$$\sum_{\omega_m \in \Omega(\Sigma_{\mathsf{ex}}^m)} f_{\Sigma_{\mathsf{ex}}^m,r_j^m}(\omega_m) \cdot \prod_{i=1,2} \alpha_i^{\Sigma_{\mathsf{ex}}^m,f_{r_i^m}(\omega_m)} = 0$$

for arbitrary indices $m \in \{1, 2, 3\}$. The quintessence of these calculations is that the values α_1 and α_2 can be determined “locally” on $\Omega(\Sigma_{\mathsf{ex}}^m)$ by evaluating just one instance of each conditional r_1 and r_2, say r_1^m and r_2^m.

From the conditional structures of the possible worlds in $\Omega(\Sigma_{\mathsf{ex}}^m)$ with respect to r_1^m and r_2^m as shown in Table 4.2, we can derive

$$\begin{aligned}
0 &= (1-p_1)\cdot\alpha_1^{1-p_1}\cdot(\alpha_2^{1-p_2}+\alpha_2^{-p_2}) - p_1\cdot\alpha_1^{-p_1}\cdot 2,\\
0 &= (1-p_2)\cdot\alpha_2^{1-p_2}\cdot(\alpha_1^{1-p_1}+1) - p_2\cdot\alpha_2^{-p_2}\cdot(\alpha_1^{1-p_1}+1).
\end{aligned}$$

The solution of this equation system is

$$\alpha_1 = \frac{2\cdot p_1}{1-p_1}\cdot(1-p_2)\cdot\left(\frac{p_2}{1-p_2}\right)^{p_2}, \qquad \alpha_2 = \frac{p_2}{1-p_2}, \tag{4.20}$$

here parameterized by the probabilities p_1 and p_2. For example, if $p_1 = 0.8$ and $p_2 = 0.5$, then $\alpha_1 = 4$ and $\alpha_2 = 1$. Note that the solution of the equation system is independent of the domain size. Here, we just fixed $|\mathsf{Const}_{\Sigma_{\mathsf{ex}}}| = 3$ in order to provide a low-threshold access to our considerations. In other examples, it can also happen that $|\mathsf{Const}_{\Sigma_{\mathsf{ex}}}|$ occurs in the solution as a parameter so that the maximum equation system can be solved analytically in dependence of the domain size $|\mathsf{Const}_{\Sigma_{\mathsf{ex}}}|$.

In a next step, this result can be used to analytically draw inferences. For example, $\mathcal{R}_{\mathsf{ex}} \models^{\mathsf{ME}}_{\Sigma_{\mathsf{ex}}} (C(X)|A(X))[p]$ holds for the following probability p:

$$\begin{aligned}
p &= \frac{\sum_{m=1}^{3} \mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{ex}},\mathcal{R}_{\mathsf{ex}}}(A(c_m)C(c_m))}{\sum_{m=1}^{3} \mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{ex}},\mathcal{R}_{\mathsf{ex}}}(A(c_m))}\\
&= \frac{\sum_{m=1}^{3}\sum_{\omega\in\Omega(\Sigma_{\mathsf{ex}}):\ \omega\models A(c_m)C(c_m)} \mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{ex}},\mathcal{R}_{\mathsf{ex}}}(\omega)}{\sum_{m=1}^{3}\sum_{\omega\in\Omega(\Sigma_{\mathsf{ex}}):\ \omega\models A(c_m)} \mathcal{P}^{\mathsf{ME}}_{\Sigma_{\mathsf{ex}},\mathcal{R}_{\mathsf{ex}}}(\omega)}\\
&= \frac{\sum_{m=1}^{3}\sum_{\omega\in\Omega(\Sigma_{\mathsf{ex}}):\ \omega\models A(c_m)C(c_m)} \prod_{i=1,2}\alpha_i^{f_{\Sigma_{\mathsf{ex}},r_i}(\omega)}}{\sum_{m=1}^{3}\sum_{\omega\in\Omega(\Sigma_{\mathsf{ex}}):\ \omega\models A(c_m)} \prod_{i=1,2}\alpha_i^{\Sigma_{\mathsf{ex}},f_{r_i}(\omega)}}\\
&= \frac{\displaystyle\sum_{m=1}^{3}\sum_{\omega_1\in\Omega(\Sigma^1_{\mathsf{ex}})}\cdots\sum_{\omega_m\in\Omega(\Sigma^m_{\mathsf{ex}}):\ \omega_m\models A(c_m)C(c_m)}\cdots\sum_{\omega_3\in\Omega(\Sigma^3_{\mathsf{ex}})}\prod_{i=1,2}\alpha_i^{\sum_{l=1}^{3} f_{\Sigma^l_{\mathsf{ex}},r^l_i}(\omega_l)}}{\displaystyle\sum_{m=1}^{3}\sum_{\omega_1\in\Omega(\Sigma^1_{\mathsf{ex}})}\cdots\sum_{\omega_m\in\Omega(\Sigma^m_{\mathsf{ex}}):\ \omega_m\models A(c_m)}\cdots\sum_{\omega_3\in\Omega(\Sigma^3_{\mathsf{ex}})}\prod_{i=1,2}\alpha_i^{\sum_{l=1}^{3} f_{\Sigma^l_{\mathsf{ex}},r^l_i}(\omega_l)}}\\
&= \frac{\displaystyle\sum_{m=1}^{3}\left(\prod_{\substack{l=1,\ldots,3:\\ l\neq m}}\sum_{\omega_l\in\Omega(\Sigma^l_{\mathsf{ex}})}\prod_{i=1,2}\alpha_i^{f_{\Sigma^l_{\mathsf{ex}},r^l_i}(\omega_l)}\right)\cdot\left(\sum_{\substack{\omega_m\in\Omega(\Sigma^m_{\mathsf{ex}}):\\ \omega_m\models A(c_m)C(c_m)}}\prod_{i=1,2}\alpha_i^{f_{\Sigma^m_{\mathsf{ex}},r^m_i}(\omega_m)}\right)}{\displaystyle\sum_{m=1}^{3}\left(\prod_{\substack{l=1,\ldots,3:\\ l\neq m}}\sum_{\omega_l\in\Omega(\Sigma^l_{\mathsf{ex}})}\prod_{i=1,2}\alpha_i^{f_{\Sigma^l_{\mathsf{ex}},r^l_i}(\omega_l)}\right)\cdot\left(\sum_{\substack{\omega_m\in\Omega(\Sigma^m_{\mathsf{ex}}):\\ \omega_m\models A(c_m)}}\prod_{i=1,2}\alpha_i^{f_{\Sigma^m_{\mathsf{ex}},r^m_i}(\omega_m)}\right)}
\end{aligned}$$

$$= \frac{3 \cdot \left(\sum\limits_{\omega_l \in \Omega(\Sigma^l_{\mathsf{ex}})} \prod\limits_{i=1,2} \alpha_i^{f_{\Sigma^l_{\mathsf{ex}}, r^l_i}(\omega_l)} \right)^{3-1} \cdot \left(\sum\limits_{\substack{\omega_m \in \Omega(\Sigma^m_{\mathsf{ex}}): \\ \omega_m \models A(c_m)C(c_m)}} \prod\limits_{i=1,2} \alpha_i^{f_{\Sigma^m_{\mathsf{ex}}, r^m_i}(\omega_m)} \right)}{3 \cdot \left(\sum\limits_{\omega_l \in \Omega(\Sigma^l_{\mathsf{ex}})} \prod\limits_{i=1,2} \alpha_i^{f_{\Sigma^l_{\mathsf{ex}}, r^l_i}(\omega_l)} \right)^{3-1} \cdot \left(\sum\limits_{\substack{\omega_m \in \Omega(\Sigma^m_{\mathsf{ex}}): \\ \omega_m \models A(c_m)}} \prod\limits_{i=1,2} \alpha_i^{f_{\Sigma^m_{\mathsf{ex}}, r^m_i}(\omega_m)} \right)}$$

$$= \frac{\sum_{\omega_m \in \Omega(\Sigma^m_{\mathsf{ex}}):\ \omega_m \models A(c_m)C(c_m)} \prod_{i=1,2} \alpha_i^{f_{\Sigma^m_{\mathsf{ex}}, r^m_i}(\omega_m)}}{\sum_{\omega_m \in \Omega(\Sigma^m_{\mathsf{ex}}):\ \omega_m \models A(c_m)} \prod_{i=1,2} \alpha_i^{f_{\Sigma^m_{\mathsf{ex}}, r^m_i}(\omega_m)}}$$

$$= \frac{\alpha_1^{1-p_1} \cdot \alpha_2^{1-p_2} + \alpha_1^{-p_1}}{\alpha_1^{1-p_1} \cdot (\alpha_2^{1-p_2} + \alpha_2^{-p_2}) + 2 \cdot \alpha_1^{-p_1}}$$

$$= \frac{\alpha_1 \cdot \alpha_2 + \alpha_2^{p_2}}{\alpha_1 \cdot (\alpha_2 + 1) + 2 \cdot \alpha_2^{p_2}}.$$

With (4.20) it follows that

$$\begin{aligned} p &= \frac{\frac{2 \cdot p_1}{1-p_1} \cdot (1-p_2) \cdot (\frac{p_2}{1-p_2})^{p_2} \cdot \frac{p_2}{1-p_2} + (\frac{p_2}{1-p_2})^{p_2}}{\frac{2 \cdot p_1}{1-p_1} \cdot (1-p_2) \cdot (\frac{p_2}{1-p_2})^{p_2} \cdot (\frac{p_2}{1-p_2} + 1) + 2 \cdot (\frac{p_2}{1-p_2})^{p_2}} \\ &= \frac{\frac{2 \cdot p_1 \cdot p_2}{1-p_1} + 1}{\frac{2 \cdot p_1 \cdot p_2}{1-p_1} + \frac{2 \cdot p_1}{1-p_1} \cdot (1-p_2) + 2} \\ &= \frac{2 \cdot p_1 \cdot p_2 + 1 - p_1}{2 \cdot p_1 \cdot p_2 + 2 \cdot p_1 - 2 \cdot p_1 \cdot p_2 + 2 \cdot (1-p_1)} \\ &= \frac{2 \cdot p_1 \cdot p_2 + 1 - p_1}{2}. \end{aligned}$$

Note that the probability p of the query conditional $(C(X)|A(X))[p]$ is independent of the domain size $|\mathsf{Const}_{\Sigma_{\mathsf{ex}}}| = 3$ as well. If $p_1 = 0.8$ and $p_2 = 0.5$, which implies $\alpha_1 = 4$ and $\alpha_2 = 1$, then $p = 0.5$, for instance.

Example 4.3.10 can be extended to an arbitrary finite size of Const_Σ. Therewith, we can lift the inference rule *transitive chaining* [Kern-Isberner, 2001a] from the propositional case to the relational setting as the next proposition states.

Proposition 4.3.11: Transitive Chaining

Let $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ be a finite signature with $|\mathsf{Const}_\Sigma| = k$ for an arbitrary $k \in \mathbb{N}$ and $\mathsf{Pred}_\Sigma = \{A/1, B/1, C/1\}$. Further, let the knowledge base $\mathcal{R} = (\emptyset, \{r_1, r_2\})$ be given by

$$r_1 = (B(X)|A(X))[p_1], \qquad r_2 = (C(X)|B(X))[p_2].$$

Then,

$$\mathcal{R} \models^{\mathsf{ME}}_{\Sigma} (C(X)|A(X))[\frac{1}{2} \cdot (2 \cdot p_1 \cdot p_2 + 1 - p_1)].$$

Proof. The proof is analogous to the argumentation in Example 4.3.10. Note that the specific number of constants in Example 4.3.10 did not play a role when computing the maximum entropy model and was just chosen to keep the equations illustrative. Here, we recall the main arguments from Example 4.3.10 for an arbitrary number $k = |\mathsf{Const}_{\Sigma}|$ of constants.

Similar to Example 4.3.10, we define the subsignatures $\Sigma^l = (\{c_l\}, \mathsf{Pred})$ as well as the instances of the conditionals r_1 and r_2 according to the schema $r_1^l = (B(c_l)|A(c_l))[p_1]$ and $r_2^l = (B(c_l)|A(c_l))[p_2]$ for $l = 1, \ldots, k$. Then, for $j = 1, 2$, we have $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}} \models r_j$ iff

$$\begin{aligned}
0 &= \sum_{\omega \in \Omega(\Sigma)} f_{\Sigma,j}(\omega) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma,i}(\omega)} \\
&= \sum_{\omega_1 \in \Omega(\Sigma^1)} \cdots \sum_{\omega_k \in \Omega(\Sigma^k)} \left(\sum_{l=1}^{k} f_{\Sigma^l, r_j^l}(\omega_l) \right) \cdot \prod_{i=1,2} \alpha_i^{\sum_{l=1}^{k} f_{\Sigma^l, r_i^l}(\omega_l)} \\
&= \sum_{\omega_1 \in \Omega(\Sigma^1)} \cdots \sum_{\omega_k \in \Omega(\Sigma^k)} \left(\sum_{l=1}^{k} f_{\Sigma^l, r_j^l}(\omega_l) \right) \cdot \prod_{i=1,2} \prod_{l=1}^{k} \alpha_i^{f_{\Sigma^l, r_i^l}(\omega_l)} \\
&= \sum_{l=1}^{k} \sum_{\omega_1 \in \Omega(\Sigma^1)} \cdots \sum_{\omega_k \in \Omega(\Sigma^k)} f_{\Sigma^l, r_j^l}(\omega_l) \cdot \prod_{i=1,2} \prod_{m=1}^{k} \alpha_i^{f_{\Sigma^m, r_i^m}(\omega_m)} \\
&= \sum_{l=1}^{k} \left(\prod_{\substack{m=1,\ldots,k: \\ m \neq l}} \sum_{\omega_m \in \Omega(\Sigma^m)} \prod_{i=1,2} \alpha_i^{f_{\Sigma^m, r_i^m}(\omega_m)} \right) \cdot \left(\sum_{\omega_l \in \Omega(\Sigma^l)} f_{\Sigma^l, r_j^l}(\omega_l) \cdot \prod_{i=1,2} \alpha_i^{f_{\Sigma^l, r_i^l}(\omega_l)} \right).
\end{aligned}$$

The instances of r_1 and r_2 are isomorphic, i.e., for both $j = 1$ and $j = 2$ it holds that, for $l, m \in \{1, \ldots, k\}$, the conditionals r_j^l and r_j^m are the same up to replacing the constant c_l in r_j^l by c_m. The same holds for the possible worlds in $\Omega(\Sigma^l)$ and $\Omega(\Sigma^m)$ which leads to the fact that the conditional structures of the possible worlds in $\Omega(\Sigma^l)$ with respect to r_j^l are the same as the conditional structures of the possible worlds in $\Omega(\Sigma^m)$ with respect to r_j^m, i.e.,

$$\sigma_{\Sigma^m, \{r_1^m, r_2^m\}}(\Omega(\Sigma^m)) = \sigma_{\Sigma^l, \{r_1^l, r_2^l\}}(\Omega(\Sigma^l)), \qquad m, l \in \{1, \ldots, k\}.$$

This again carries over to the feature functions f_{Σ^l, r_j^l} and f_{Σ^m, r_j^m} which means that

$$\sum_{\omega_l \in \Omega(\Sigma^l)} \prod_{i=1,2} \alpha_i^{f_{\Sigma^l, r_i^l}(\omega_l)} = \sum_{\omega_m \in \Omega(\Sigma^m)} \prod_{i=1,2} \alpha_i^{f_{\Sigma^m, r_i^m}(\omega_m)}$$

as well as

$$\sum_{\omega_l\in\Omega(\Sigma^l)} f_{\Sigma^l,r_j^l}(\omega_l)\cdot\prod_{i=1,2}\alpha_i^{f_{\Sigma^l,r_i^l}(\omega_l)} = \sum_{\omega_m\in\Omega(\Sigma^m)} f_{\Sigma^m,r_j^m}(\omega_m)\cdot\prod_{i=1,2}\alpha_i^{f_{\Sigma^m,r_i^m}(\omega_m)}$$

hold for $l, m \in \{1,\ldots,k\}$ and $j = 1, 2$ just as in Example (4.3.10). Consequently, we have (cf. Example (4.3.10))

$$\begin{aligned}
0 &= \sum_{l=1}^{k}\left(\prod_{\substack{m=1,\ldots,k:\\ m\neq l}}\ \sum_{\omega_m\in\Omega(\Sigma^m)}\prod_{i=1,2}\alpha_i^{f_{\Sigma^m,r_i^m}(\omega_m)}\right)\cdot\left(\sum_{\omega_l\in\Omega(\Sigma^l)} f_{\Sigma^l,r_j^l}(\omega_l)\cdot\prod_{i=1,2}\alpha_i^{f_{\Sigma^l,r_i^l}(\omega_l)}\right)\\
&= \underbrace{k\cdot\left(\sum_{\omega_m\in\Omega(\Sigma^m)}\prod_{i=1,2}\alpha_i^{f_{\Sigma^m,r_i^m}(\omega_m)}\right)^{k-1}}_{>0}\cdot\left(\sum_{\omega_m\in\Omega(\Sigma^m)} f_{\Sigma^m,r_j^m}(\omega_m)\cdot\prod_{i=1,2}\alpha_i^{f_{\Sigma^m,r_i^m}(\omega_m)}\right)\\
&= \sum_{\omega_m\in\Omega(\Sigma^m)} f_{\Sigma^m,r_j^m}(\omega_m)\cdot\prod_{i=1,2}\alpha_i^{f_{\Sigma^m,r_i^m}(\omega_m)}
\end{aligned}$$

for an arbitrary index $m \in \{1,\ldots,k\}$. As we have reduced the maximum entropy equation system of $\mathcal{R}$ with respect to Σ with $|\mathsf{Const}_\Sigma| = k$ to the respective maximum entropy equation system of $\mathcal{R}$ with respect to Σ^m and $|\mathsf{Const}_{\Sigma^m}| = 1$, we are exactly in the same situation as in Example 4.3.10 after the respective reformulation steps. Hence, as a solution of the equation system, we obtain (4.20) and, consequently, it is $p = \frac{1}{2}\cdot(2\cdot p_1\cdot p_2 + 1 - p_1)$. □

With similar arguments as used in the proof of Proposition 4.3.11, we can generalize many further inference rules. Here, we give three additional examples (cf. [Kern-Isberner, 2001a] for their propositional variants).

Proposition 4.3.12: Lifted Inference Rules
Let $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ be a finite signature with $|\mathsf{Const}_\Sigma| = k$ where $k \in \mathbb{N}$ and $\mathsf{Pred}_\Sigma = \{A/1, B/1, C/1\}$.

- Categorial Specificity:
  Let $\mathcal{R} = (\{F_1\}, \{r_1, r_2\})$ be a knowledge base with the fact $F_1 = \forall X.C(X) \Rightarrow A(X)$ and the conditionals $r_1 = (B(X)|A(X))[p_1]$ and $r_2 = (B(X)|C(X))[p_2]$ where $p_1, p_2 \in (0, 1)$. Then,

$$\mathcal{R} \models_{\Sigma}^{\mathsf{ME}} (B(X)|A(X)C(X))[p_2].$$

- Cautious Monotonicity:
  Let $\mathcal{R} = (\emptyset, \{r_1, r_2\})$ be a knowledge base with the conditionals $r_1 = (B(X)|A(X))[p_1]$ and $r_2 = (C(X)|A(X))[p_2]$ where $p_1, p_2 \in (0,1)$. Then,
  $$\mathcal{R} \models^{\mathsf{ME}}_{\Sigma} (C(X)|A(X)B(X))[p_2].$$
- Cautious Cut:
  Let $\mathcal{R} = (\emptyset, \{r_1, r_2\})$ be a knowledge base with the conditionals $r_1 = (C(X)|A(X)B(X))[p_1]$ and $r_2 = (B(X)|A(X))[p_2]$ where $p_1, p_2 \in (0,1)$. Then,
  $$\mathcal{R} \models^{\mathsf{ME}}_{\Sigma} (C(X)|A(X))[\frac{1}{2}(2 \cdot p_1 \cdot p_2 + 1 - p_2)].$$

Proof (Sketch). With the same argumentation as in the proof of Proposition 4.3.11, the computations of the maximum entropy models can be “propositionalized”, which means a reduction to the case where the signature mentions only one constant here. The important prerequisite for this, which holds in all cases—*categorial specificity*, *cautious monotony*, and *cautious cut*—is that the instances of the conditionals in the respective knowledge bases syntactically split over $\Sigma^1, \ldots, \Sigma^k$ where $\Sigma^l = (\{c_l\}, \mathsf{Pred}_\Sigma)$ for $l \in \{1, \ldots, k\}$. This means that the instances of each conditional mention one (distinct) constant only. Also, the fact $F_1 = \forall X.C(X) \Rightarrow A(X)$ from the knowledge base in the case of *categorial specificity* can be replaced by the set of facts $\{C(c_l) \Rightarrow A(c_l) \mid c_l \in \mathsf{Const}_\Sigma\}$ and the maximum entropy equation system of $\mathcal{R} = (\{F_1\}, \{r_1, r_2\})$ with respect to Σ reduced to

$$\mathcal{R}^l = (\{C(c_l) \Rightarrow A(c_l)\}, \{(B(c_l)|A(c_l))[p_1], (B(c_l)|C(c_l))[p_2]\})$$

with respect to Σ^l. Once this reduction is done, the inferences mentioned in this proposition directly follow from the respective results in the propositional case (cf. [Kern-Isberner, 2001a]). ☐

In this section, we have shown that the Σ-ME-vector of a knowledge base $\mathcal{R}$ can sometimes be calculated “locally” by splitting the possible worlds in $\Omega(\Sigma)$ into partial possible worlds and by evaluating instances of the expressions (facts and conditionals) in $\mathcal{R}$ independently with respect to these partial possible worlds. This is especially effective when the instances are isomorphic, i.e., when they are evaluated the same. Then, only one instance per expression has to be considered and the result can be generalized to the other instances like in Example 4.3.10. In [Wilhelm et al., 2017b], we have applied these strategies to arbitrary knowledge bases where the logical parts are defined by using Boolean combinations of unary predicates. Hence, this result lifts Proposition 3.2.5 to the probabilistic case. We recall the respective proposition from [Wilhelm et al., 2017b].

Proposition 4.3.13: Maximum Entropy Model and Unary Predicates *(cf. [Wilhelm et al., 2017b])*

Let $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ be a finite signature. Suppose $|\mathsf{Const}_\Sigma| \geq 1$ and Pred_Σ being a set of unary predicates. Further, let $\mathcal{R} = (\emptyset, \mathcal{B}_\mathcal{R})$ be a Σ-p-consistent knowledge base with

$$\mathcal{B}_\mathcal{R} = \{(\psi_1(X)|\phi_1(X)[p_1], \ldots, (\psi_n(X)|\phi_n(X)[p_n]\},$$

where ϕ_i and ψ_i, for $i = 1, \ldots, n$, are Boolean combinations of unary predicates from Pred_Σ. Further, let ϕ and ψ be Boolean combinations of unary predicates from Pred_Σ as well. Then,

$$\mathcal{R} \models^{\mathsf{ME}}_{\Sigma} (\psi(X)|\phi(X))[p] \quad \text{iff} \quad \mathcal{R}^c \models^{\mathsf{ME}}_{\Sigma_c} (\psi(c)|\phi(c))[p]$$

where $\Sigma_c = (\{c\}, \mathsf{Pred}_\Sigma)$ for any $c \in \mathsf{Const}_\Sigma$ and $\mathcal{R}^c = (\emptyset, \mathcal{B}^c_\mathcal{R})$ with

$$\mathcal{B}^c_\mathcal{R} = \{(\psi_1(c)|\phi_1(c)[p_1], \ldots, (\psi_n(c)|\phi_n(c)[p_n]\}.$$

Proof Idea. The proof is analogous to the proof of Proposition 4.3.11. ☐

In Chapter 6, we will generalize this line of research and assemble our strategies to a more sophisticated and more general framework for simplifying the maximum entropy equation system (4.19) which we call *typed model counting*. Even in cases where the maximum entropy equation system cannot be solved analytically, these simplifications are essential. The maximum entropy equation system (4.19) as well as the objective function of the dual maximum entropy optimization problem (4.12) involve sums over all potentially positive possible worlds. As the amount of the possible worlds is exponential in the domain size k, it is not possible to develop fast numerical solvers for the maximum entropy problem without a compact representation of these sums over possible worlds. With typed model counting we will provide a formal framework with which it is possible to overcome this obstacle of the exponential dependency on k for some classes of knowledge bases that go beyond knowledge bases solely based on unary predicates.

Before we present our concept of typed model counting in Chapter 6, we insert an excursion on the connection between the principle of maximum entropy and *Markov Logic Networks* [Richardson and Domingos, 2006] (Section 4.4) as well as a chapter on *lifted inference* (Chapter 5). In Chapter 5, we will substantiate the task of drawing inferences at maximum entropy and formalize the dependency of this task on the domain size k. We will also discuss with *condensed iterative scaling* in Section 5.2 the numerical basis for our investigations on efficient maximum entropy calculations.

4.4 Maximum Entropy and Markov Logic Networks

In this short excursion, we establish a connection between maximum entropy models of Σ-p-consistent knowledge bases $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ and *Markov Logic Networks*. *Markov Logic Networks* (MLNs, [Richardson and Domingos, 2006]) constitute a popular approach in the field of statistical relational learning (cf. [Getoor and Taskar, 2007]) by combining probabilistic graphical models, namely *Markov Random Fields*, and first-order logic. For MLNs there exist well-investigated techniques for both exact and approximative inference (cf. [Beedkar et al., 2013] and, in particular, [Van Haaren et al., 2016] as well as the references cited there). Here, we show that the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ of $\mathcal{R}$ can be compiled into an MLN and, therewith, one can principally benefit from the inference techniques for MLNs.

Definition 4.4.1: Markov Logic Network

(based on [Richardson and Domingos, 2006])

Let Σ be a finite signature. A *Markov Logic Network* $\mathcal{M}$ is a set of pairs (F_i, ν_i) consisting of formulas $F_i \in \mathcal{RL}(\Sigma)$ and weights $\nu_i \in \mathbb{R}$ which defines a probability distribution $\mathcal{P}_{\mathcal{L}} : \Omega(\Sigma) \to [0,1]$ by

$$\mathcal{P}_{\mathcal{M}}(\omega) = \frac{1}{\zeta} \cdot \exp\left(\sum_i \nu_i \cdot \mathsf{cnt}(F_i, \omega)\right),$$

where ζ is a normalization constant and

$$\mathsf{cnt}(F_i, \omega) = |\{F_i' \in \mathsf{Inst}_\Sigma(F_i) \mid \omega \models_\Sigma F_i'\}|$$

is the number of instances of F_i that are Σ-satisfied in the possible world $\omega \in \Omega(\Sigma)$.

Note that in order to represent hard constraints, i.e., factual knowledge, it is convenient to admit infinite weights in MLNs as well. We have the following proposition.

Proposition 4.4.2: Maximum Entropy and Markov Logic Networks

(cf. [Wilhelm et al., 2019b])

Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a Σ-p-consistent knowledge base, let $\vec{\alpha}_{\Sigma,\mathcal{R}} = (\alpha_1, \ldots, \alpha_n)$ be a ME-vector of $\mathcal{R}$ wrt. Σ, let α_0 be the respective normalizing constant, and let $\mathcal{M}$ be the MLN defined by

- $(\phi_i\psi_i, (1-p_i)\cdot\log\alpha_i) \in \mathcal{M}$ for all $r_i = (\psi_i|\phi_i)[p_i]$ from $\mathcal{B}_\mathcal{R}$,
- $(\phi_i\overline{\psi_i}, -p_i\cdot\log\alpha_i) \in \mathcal{M}$ for all $r_i = (\psi_i|\phi_i)[p_i]$ from $\mathcal{B}_\mathcal{R}$,
- and $(\overline{F}, -\infty) \in \mathcal{M}$ for all $F \in \mathcal{F}_\mathcal{R}$.

Then, $\mathcal{P}_\mathcal{M} = \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$.

Proof. Let $\omega \in \Omega^0_{\mathcal{F}_\mathcal{R}}(\Sigma) = \Omega(\Sigma)\setminus\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$. Then, there is at least one $F' \in \mathcal{F}_\mathcal{R}$ with $\omega \not\models_\Sigma F'$ but $\omega \models_\Sigma \overline{F'}$. Hence, $\mathsf{cnt}(\overline{F'},\omega) = 1$ (note that $\overline{F'}$ is a sentence and, thus, $\mathsf{Inst}_\Sigma(\overline{F'}) = \{\overline{F'}\}$ holds which means that $\mathsf{cnt}(\overline{F'},\omega) \leq 1$), and

$$\begin{aligned}\mathcal{P}_\mathcal{M}(\omega) &= \lim_{\nu\to-\infty}\frac{1}{\zeta}\cdot\exp\left(\sum_{F\in\mathcal{F}_\mathcal{R}}\nu\cdot\mathsf{cnt}(\overline{F},\omega) + \sum_{r_i\in\mathcal{B}_\mathcal{R}}(1-p_i)\cdot\log(\alpha_i)\cdot\mathsf{cnt}(\phi_i\psi_i,\omega)\right.\\ &\qquad\left.+\sum_{r_i\in\mathcal{B}_\mathcal{R}}-p_i\cdot\log(\alpha_i)\cdot\mathsf{cnt}(\phi_i\overline{\psi_i},\omega)\right)\\ &= \lim_{\nu\to-\infty}\frac{1}{\zeta}\cdot\exp(\nu\cdot\mathsf{cnt}(\overline{F'},\omega))\\ &= \frac{1}{\zeta}\cdot\lim_{\nu\to-\infty}\exp(\nu)\\ &= 0 = \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega).\end{aligned}$$

Now, let $\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)$. Then, $\mathsf{cnt}(\overline{F},\omega) = 0$ for all $F \in \mathcal{F}_\mathcal{R}$, and

$$\begin{aligned}&\mathcal{P}_\mathcal{M}(\omega)\\ &= \frac{1}{\zeta}\cdot\exp\left(\sum_{r_i\in\mathcal{B}_\mathcal{R}}(1-p_i)\cdot\log(\alpha_i)\cdot\mathsf{cnt}(\phi_i\psi_i,\omega) + \sum_{r_i\in\mathcal{B}_\mathcal{R}}-p_i\cdot\log(\alpha_i)\cdot\mathsf{cnt}(\phi_i\overline{\psi_i},\omega)\right)\\ &= \frac{1}{\zeta}\cdot\prod_{i=1}^n\alpha_i^{(1-p_i)\cdot\mathsf{cnt}(\phi_i\psi_i,\omega)}\cdot\prod_{i=1}^n\alpha_i^{-p_i\cdot\mathsf{cnt}(\phi_i\overline{\psi_i},\omega)}\\ &= \frac{1}{\zeta}\cdot\prod_{i=1}^n\alpha_i^{\mathsf{cnt}(\phi_i\psi_i,\omega)-p_i\cdot(\mathsf{cnt}(\phi_i\psi_i,\omega)+\mathsf{cnt}(\phi_i\overline{\psi_i},\omega))}\\ &= \frac{1}{\zeta}\cdot\prod_{i=1}^n\alpha_i^{\mathsf{ver}_{\Sigma,r_i}(\omega)-p_i\cdot\mathsf{app}_{\Sigma,r_i}(\omega)} = \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega)\end{aligned}$$

with the normalizing constant $\zeta = \alpha_0$. □

Note that Proposition 4.4.2 gives the weights of Markov Logic Networks a profound meaning. The standard way of coming up with the weights of MLNs is to learn them from data sets which does not allow for a direct and clear interpretation of the weights in every case. A more in-depth analysis of the connection between maximum entropy and Markov Logic Networks remains future work.

5 Towards Lifted Inference at Maximum Entropy

A general problem of probabilistic reasoning in relational settings is that the underlying probability spaces grow exponentially in the domain size. Thus, with *statistical relational AI* [De Raedt et al., 2016; Getoor and Taskar, 2007] a whole research field has emerged which deals with the question under what circumstances probabilistic inferences can be drawn in a domain-lifted way. With *domain-lifted* one typically means that inferences can be drawn in time polynomial in the domain size. In this chapter, we discuss what domain-lifted inference means for maximum entropy reasoning (Section 5.1), and—because existing approaches are not able to deal with lifted inference at maximum entropy—we present with *condensed iterative scaling* a novel algorithm for computing the maximum entropy vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$ and, therewith, maximum entropy probabilities (Section 5.2).

5.1 Lifted Inference

Drawing *domain-lifted inferences*, or *lifted inferences* for short, is *the* major challenge in *statistical relational artificial intelligence* (StarAI) [De Raedt et al., 2016], also known as *statistical relational learning* (SRL) [Getoor and Taskar, 2007], which is the subdiscipline of artificial intelligence that deals with probabilistic reasoning in complex relational settings. In literature, the term *domain-lifted inference* is not used uniformly and rather informally. The common fundamental idea is to address the task of drawing inference in a tractable amount of time even when the size of the considered domain is very large.

The domain-liftability of an inference problem is of great importance because runtimes of non-domain-liftable inferences become intractable with growing domain size quite fast. This is because exact inferences require to sum up the probabilities of all relevant possible worlds which are exponentially many when measured in the domain size. In this section, we collect different conceptualizations of *lifted inference* from literature and formally define *domain-lifted inference at maximum entropy.*

▶ Conceptualization of Lifted Inference

The probably most formal definition of *domain-lifted inference* comes from the field of complexity theory which identifies domain-lifted inference as a generic term used to subsume all kinds of inferences that can be drawn in time polynomial in the domain size, here in $k = |\mathsf{Const}_\Sigma|$ (cf. e.g., [Van den Broeck, 2013]). However, besides this rather technical interpretation, there are paraphrases of the term *domain-liftability* which focus on more conceptual aspects, too. For instance, the authors of [Apsel and Brafman, 2011] call drawing lifted inferences "the act of exploiting the high level structure in relational models."

Other explanations can be understood as instructions on how domain-lifted inference can be implemented. Kisynski and Poole note that "the idea behind lifted inference is to carry out as much inference as possible without propositionalizing" [Kisynski and Poole, 2009]. In our setting, propositionalization can be understood as breaking down conditionals to the level of ground instances. That is, lifted inference means to avoid grounding as far as possible. In contrast to the evaluation of individual instances, "lifted inference [deals] with groups of indistinguishable variables" [Singla et al., 2010].

Basically, domain-liftability can be achieved in two ways:

1. By applying elaborated, problem-adjusted solving techniques to the inference task. These solving techniques typically exploit combinatorial arguments to reduce the complexity of the problem.

2. By restricting the class of the considered inference problems to those which are domain-liftable. This, however, means a loss of expressivity when formulating knowledge bases and/or queries.

▶ Lifted Inference at Maximum Entropy

In the context of maximum entropy reasoning, a formal definition of domain-lifted inference has to take into account that computing the maximum entropy model means solving a non-linear optimization problem which cannot be solved analytically in general but requires approximate methods. Thus, we identify the goal of domain-lifted maximum entropy reasoning as follows: First, we want to determine inference problems for which there is an approximation scheme which produces an approximation to the maximum entropy model of appropriate quality in time polynomial in the domain size k. Then, we want to draw exact inferences from this approximation, again in polynomial time.

Because the number of possible worlds is exponential in k, we cannot compute and store the approximation of the maximum entropy model on the level of possible worlds. Instead, we aim at calculating an approximation $\vec{\alpha}^*_{\Sigma,\mathcal{R}}$ of the Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$ (cf. Definition 4.3.1) for a given knowledge base $\mathcal{R}$. Recall that the size of such an approximation vector $\vec{\alpha}^*_{\Sigma,\mathcal{R}}$ is constant in k. Then, we draw exact inferences based on the probability distribution $\mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}$ by computing only those probability values $\mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}(\omega)$ which are needed for answering the query.

Definition 5.1.1: Domain-Lifted Maximum Entropy Inference
Let Σ be a finite signature, let $\mathcal{R} \subseteq \mathcal{RPCL}(\Sigma)$ be a Σ-p-consistent knowledge base where the probabilities of the conditionals in $\mathcal{R}$ are given as rational numbers, and let $\delta \in \mathcal{RCL}(\Sigma)$ be a qualitative relational conditional. We call the maximum entropy inference task $\mathcal{R} \models^{\mathsf{ME}}_{\Sigma} \delta[?]$ (cf. Definitions 4.1.16 and 4.2.13) *domain-liftable* iff

- there is an algorithm which produces an approximation $\vec{\alpha}^*_{\Sigma,\mathcal{R}}$ of the Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$ with a prespecified error tolerance in time polynomial in the domain size $k = |\mathsf{Const}_{\Sigma}|$,
- and $\mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}(\delta)$ can be computed in time polynomial in k.

Definition 5.1.1 addresses the computation problem of the maximum entropy inference task (cf. also Definition 4.1.16 for the definition of the computation and decision problem of probabilistic inferences in general). We will focus on this computation problem in the following. In order to tackle the decision problem of the maximum entropy inference task, i.e., the question whether $\mathcal{R} \models^{\mathsf{ME}}_{\Sigma} \delta[p]$ holds for a given rational number $p \in [0,1]$, we proceed as follows. We compute an approximation $\vec{\alpha}^*_{\Sigma,\mathcal{R}}$ of the Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$ as in Definition 5.1.1 as well as the probability $p' \in [0,1]$ for which $\mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}(\delta) = p'$ holds. That is, we solve the computation problem first. Then, we decide whether $p \in [p' - \varepsilon, p' + \varepsilon]$ holds where $\varepsilon > 0$ is a prespecified precision. Note that there are a few cases in which the decision problem can be solved directly without calculating an approximation to the Σ-ME-vector or the maximum entropy Σ-model (cf. Example 4.3.10). Also note that Proposition 4.3.13 proves that lifted inference at maximum entropy is possible at least for specific subclasses of knowledge bases.

In the next section, we will discuss with *condensed iterative scaling* (CIS) [Wilhelm et al., 2019b] an algorithm for approximating the Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$ by solving the dual maximum entropy optimization problem (cf. Definition 4.3.1). As we will see, the main challenge of approximating $\vec{\alpha}_{\Sigma,\mathcal{R}}$ in a domain-lifted manner is to represent the objective function of the dual maximum entropy optimization problem so compactly that it can be stored and processed in time polynomial in k. The condensed iterative scaling algorithm CIS is adapted to this task.

5.2 Condensed Iterative Scaling

In order to solve the dual maximum entropy optimization problem (4.12) for a Σ-p-consistent knowledge base $\mathcal{R}$, we enhance the *general iterative scaling* algorithm from [Darroch and Ratcliff, 1972] to *condensed iterative scaling*. The benefit of *condensed iterative scaling* is that it is an algorithm highly adapted to the task of approximating the maximum entropy vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$ under the aggregating semantics.

In this section, we summarize the techniques for computing the maximum entropy model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ which we have discussed in Chapter 4 in order to motivate the need for a problem-adapted solving technique. Afterwards, we discuss *generalize iterative scaling* in the representation of [Finthammer, 2012] as the starting point for our subsequent developing of *condensed iterative scaling* for approximating $\vec{\alpha}_{\Sigma,\mathcal{R}}$ and, therewith, $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$.

▶ Motivation of Condensed Iterative Scaling

In Chapter 4, we have discussed three possible approaches to compute the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ of a Σ-p-consistent knowledge base $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ which we briefly recall here. For this, suppose $\mathcal{B}_{\mathcal{R}} = \{r_1, \ldots, r_n\}$.

1. **Primal Maximum Entropy Optimization Problem (4.10):**

$$\begin{aligned} &\text{minimize} \sum_{\omega \in \Omega(\Sigma)} \mathcal{P}(\omega) \cdot \log(\mathcal{P}(\omega)) \\ &\text{subject to} \\ \sum_{\omega \in \Omega(\Sigma)} f_{\Sigma,i}(\omega) \cdot \mathcal{P}(\omega) &= 0, \qquad i = 1, \ldots, n, \\ \sum_{\omega \in \Omega(\Sigma)} \mathcal{P}(\omega) &= 1, \\ \mathcal{P}(\omega) &= 0, \qquad \forall \omega \in \Omega^0_{\mathcal{F}_{\mathcal{R}}}(\Sigma), \\ -\mathcal{P}(\omega) &\leq 0, \qquad \forall \omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma), \end{aligned}$$

2. **Dual Maximum Entropy Optimization Problem (4.12):**

$$\begin{aligned} \text{minimize} \qquad & \sum_{\omega \in \Omega_{\mathcal{F}}(\Sigma)} \prod_{i=1}^{n} x_i^{f_{\Sigma,i}(\omega)} \\ \text{subject to} \qquad & x_i > 0, \qquad i = 1, \ldots, n, \end{aligned}$$

and use the solution $\vec{\alpha}_{\Sigma,\mathcal{R}} = (\alpha_1, \ldots, \alpha_n)$ to compute

$$\alpha_0 = \left(\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \prod_{i=1}^{n} \alpha_i^{f_{\Sigma,i}(\omega)} \right)^{-1}$$

and then

$$\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) = \begin{cases} \alpha_0 \cdot \prod_{i=1}^{n} \alpha_i^{f_{\Sigma,i}(\omega)} & \text{if} \quad \omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma) \\ 0 & \text{otherwise} \end{cases},$$

3. **Maximum Entropy Equation System (4.19):**

$$\text{solve} \quad \sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} f_{\Sigma,j}(\omega) \cdot \prod_{i=1}^{n} x_i^{f_{\Sigma,i}(\omega)} = 0, \qquad j = 1, \ldots, n,$$

   and use the solution $\vec{\alpha}_{\mathcal{R},\Sigma}$ to compute α_0 and then $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ in the same way as described in the case of the dual maximum entropy problem.

All three approaches have in common that they cannot be solved analytically in general but require numerical solving techniques. Being a convex optimization problem, the numerical behavior of the maximum entropy optimization problem (4.10) is quite well understood. This also holds for the dual maximum entropy optimization problem (4.12) which is—up to the substitution $\alpha_i = \exp(-\lambda_i)$ for $i = 1, \ldots, n$ where the λ_i's are the Lagrangian multipliers—also a convex optimization problem. A vast number of algorithms which solve these kinds of problems have been developed (see [Malouf, 2002] for a discussion of solvers for the maximum entropy optimization problems). Two early algorithms among them are *generalized iterative scaling* (GIS) [Darroch and Ratcliff, 1972] and *improved iterative scaling* (IIS) [Berger et al., 1996; Della Pietra et al., 1997] which can be used to solve the primal and the dual maximum entropy optimization problem, respectively.

Besides the iterative scaling methods, *gradient descent methods* (cf. [Chong and Zak, 2013]) constitute another nowadays widely used class of solution methods which can be applied to the dual maximum entropy optimization problem (4.12) as well. Moreover, since the maximum entropy model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ can also be deduced from the solution of the maximum entropy equation system (4.19), it is expedient to apply root-finding algorithms like Newton's method to (4.19) in order to obtain $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$, too.

However, all these methods have the same obstacle in common: The computational costs when applying straightforward implementations of these methods to any of the three formulations of the maximum entropy problem are exponential in the domain size $k = |\mathsf{Const}_\Sigma|$ because all of them involve iterations over all possible worlds at some point. Note that the objective functions of the optimization problems respectively the equations in the maximum entropy equation system involve sums over possible worlds which are the bottlenecks of the algorithms. The only expedient to this problem is to develop problem-adapted solvers which can handle reformulations of these sums.

Because the primal optimization problem has a solution vector of size $|\Omega(\Sigma)|$, the exponential dependency on k can hardly be overcome when solving the primal problem. Only in some rare cases it is possible to reduce computational costs substantially, e.g., by the transition from possible worlds to equivalence classes of possible worlds with respect to their conditional structure [Finthammer, 2012] or by exploiting independencies [Wilhelm et al., 2018]. In contrast to that, both the dual optimization problem and the maximum entropy equation system mention $n = |\mathcal{B}_{\mathcal{R}}|$ variables only which is constant in k and, hence, more promising. Thus, there is hope to find solution techniques for these problems which can get rid of the exponential dependency on k. As said, such solution techniques have to adapt to the concrete problem setting as either the objective function of the dual problem or the maximum entropy equation system has to be represented compactly and processed while maintaining this compact representation. Therefore, the solution techniques cannot be found among the generic solvers for convex optimization problems or the common root-fining algorithms in literature. Our goal in this thesis is to develop with *condensed iterative scaling* and *typed model counting* (cf. Chapter 6) an intertwining tandem of techniques for efficiently solving the dual maximum entropy problem. The idea is that with typed model counting we compute compact representations of the sums over possible worlds and with condensed iterative scaling we have an optimization algorithm which can deal with these representations.

Although it has been shown in experiments that gradient descend methods outperform the existing iterative scaling methods [Malouf, 2002], we focus on iterative scaling as the basic method for computing the maximum entropy model here. There are two main reasons for this: First, there has already been some effort in optimizing iterative scaling methods to solve the maximum entropy problem in combination with the aggregating semantics [Finthammer, 2012; Finthammer and Beierle, 2014] and, hence, we can build upon the results here. And second, we develop with *condensed iterative scaling* a variant of iterative scaling which, as mentioned above, solves the dual maximum entropy optimization problem based on a highly condensed input and, therewith, overcomes the disadvantage of iterating over all possible worlds in $\Omega(\Sigma)$ completely, at least for many input knowledge bases, which presumably has more influence on the computational costs than the decision whether to choose iterative scaling or gradient descent methods. It is to be expected that similar techniques for dealing with the domain size k, as we use in condensed iterative scaling, will also work for gradient descent methods.

▶ Generalized Iterative Scaling

We recall the *generalized iterative scaling (*GIS*)* algorithm from [Finthammer, 2012] and discuss its improvements made in [Finthammer and Beierle, 2014] as the status quo of numerically solving the maximum entropy optimization problem under the aggregating semantics. It will constitute the basis from which we develop *condensed iterative scaling*.

The GIS-algorithm in [Finthammer, 2012] is already an adaption of the generalized iterative scaling algorithm from [Darroch and Ratcliff, 1972]. While the latter fits any *log-linear model* (cf. [Christensen, 1997]), the former is tailor-made for approximating the maximum entropy model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ under the aggregating semantics. The pseudocode of GIS is recalled in a slightly modified way in Figure 5.1. The difference to the GIS-algorithm in [Finthammer, 2012] is that knowledge bases in [Finthammer, 2012] do not mention facts but consist of non-factual conditionals only. Non-factual conditionals are called *soft conditionals* in [Finthammer, 2012]. The integration of facts does not affect the validity of the algorithm, though, as we have already discussed that it means a transformation of the probability space only.

The basic idea of GIS is to calculate a sequence of probability distributions which converges against $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$. This is done by iteratively minimizing the relative entropy, also known as the *Kulback-Leibler divergence* (see Definition A.1.7 in the appendix for a formal definition), between the current model $\mathcal{P}$ of $\mathcal{R}$ and the uniform distribution $\mathcal{P}^{u}_{\Sigma,\mathcal{R}}$ on $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ (cf. (4.8)). This procedure converges against $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ because of the well-known identity (cf. [Csiszar, 1989])

$$\arg \min_{\mathcal{P}\in\mathsf{probMod}_{\Sigma}(\mathcal{R})} \mathcal{KL}(\mathcal{P},\mathcal{P}^{u}_{\Sigma,\mathcal{R}}) = \arg \max_{\mathcal{P}\in\mathsf{probMod}_{\Sigma}(\mathcal{R})} \mathcal{H}(\mathcal{P}),$$

where $\mathcal{KL}(\mathcal{P},\mathcal{P}^{u}_{\Sigma,\mathcal{R}})$ is the Kulback-Leibler-divergence of $\mathcal{P}$ and $\mathcal{P}^{u}_{\Sigma,\mathcal{R}}$. In [Finthammer, 2012], it is shown that the GIS-algorithm is sound and complete when the input knowledge base is Σ-p-consistent. We discuss the working of the GIS-algorithm in more detail now.

The GIS-algorithm starts with the uniform distribution $\mathcal{P}^{u}_{\Sigma,\mathcal{R}}$ (lines 2-5) and a precalculation of constants which are used in the subsequent iteration process. These constants are the normalized feature functions $\tilde{f}_{\Sigma,i}(\omega)$ for the conditionals r_i in $\mathcal{B}_{\mathcal{R}}$ and all positive possible worlds ω (cf. (3.14)), including a correctional term $\tilde{f}_{\Sigma,0}(\omega)$. The correctional term is necessary because the normalized instead of the ordinary feature functions are used (lines 8-11). In addition, the weights μ_i are derived from the probabilities and the number of instances of the conditionals in $\mathcal{B}_{\mathcal{R}}$ (lines 12-14). Formally, these weights are the expectation values of the normalized feature functions with respect to the initial uniform distribution. We call them the *initial expectation values*.

The main part of the algorithm is the ensuing iteration process in which the

Input: Σ-p-consistent knowledge base $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ with $\mathcal{B}_{\mathcal{R}} = \{r_1, \ldots, r_n\}$
and $r_i = (\psi_i|\phi_i)_{\mathsf{CS}_i}[p_i]$ for $i = 1, \ldots, n$, a termination criterion
Output: Approximation $\mathcal{P}^*$ of the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\mathcal{R},\Sigma}$

```
# Initialize with uniform distribution on $\Omega_{\mathcal{F}_{\mathcal{R}}}(\omega)$
FOR $\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$:
  $\mathcal{P}_{(0)}(\omega) = |\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)|^{-1}$
FOR $\omega \in \Omega_{\mathcal{F}^0_{\mathcal{R}}}(\Sigma)$:
  $\mathcal{P}^*(\omega) = 0$

# Precalculate constants
FOR $\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$:
  FOR $i = 1, \ldots, n$:
    Calculate normalized feature functions $\tilde{f}_{\Sigma,i}(\omega)$ (cf. (3.14))
  $\tilde{f}_{\Sigma,0}(\omega) = 1 - \sum_{i=1}^{n} \tilde{f}_{\Sigma,i}(\omega)$
FOR $i = 1, \ldots, n$:
  $\mu_i = p_i \cdot |\mathsf{Inst}_{\Sigma}(r_i)| \cdot (\sum_{j=1}^{n} |\mathsf{Inst}_{\Sigma}(r_j)|)^{-1}$
$\mu_0 = 1 - \sum_{i=1}^{n} \mu_i$

# Initialize iteration counter
$k = 0$

# Iteratively adjust probabilities
REPEAT
  $k = k + 1$
  FOR $i = 0, 1, \ldots, n$:
    $\eta_{(k),i} = \mu_i \cdot (\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}_{k-1}(\omega) \cdot \tilde{f}_{\Sigma,i}(\omega))^{-1}$   # Calculate scaling factor
  FOR $\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$:
    $\mathcal{P}'_k(\omega) = \mathcal{P}_{k-1}(\omega) \cdot \prod_{i=0}^{n} (\eta_{(k),i})^{\tilde{f}_{\Sigma,i}(\omega)}$   # Scale probabilities
  FOR $\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$:
    $\mathcal{P}_k(\omega) = \mathcal{P}'_k(\omega) \cdot (\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}'_k(\omega))^{-1}$   # Normalize probabilities
UNTIL $\langle$termination condition$\rangle$

# Return approximation $\mathcal{P}^*$
FOR $\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\omega)$:
  $\mathcal{P}^*(\omega) = \mathcal{P}_k(\omega)$
RETURN $P^*$
```

Algorithm 5.1: Algorithm GIS which returns an approximation $\mathcal{P}^*$ of $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ for a given Σ-p-consistent knowledge base $\mathcal{R}$ (adaption of GIS from [Finthammer, 2012]).

uniform distribution $\mathcal{P}^{u}_{\Sigma,\mathcal{R}}$ is iteratively scaled. This scaling process happens in lines 24-25 where, in the k-th iteration step, the current probability $\mathcal{P}_{k-1}(\omega)$ is multiplied with the scaling factor $\prod_{i=0}^{n}(\eta_{(k),i})^{\tilde{f}_{\Sigma,i}(\omega)}$ in which $\eta_{(k),i}$ is the quotient of the initial expectation value ε_i of the i-th normalized feature function $\tilde{f}_{\Sigma,i}$ and the current expectation value, i.e., the expectation value of the normalized feature function $\tilde{f}_{\Sigma,i}$ with respect to the current probability distribution $\mathcal{P}_{k-1}$ (lines 22-23). Afterwards, the resulting new expression $\mathcal{P}'_k$ has to be normalized in order to obtain a probability distribution (lines 26-27).

By repeating this scaling process with increasing iteration counter k one obtains a sequence of probability distributions $(\mathcal{P}_k)_{k\in\mathbb{N}}$ which converges against $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ [Finthammer, 2012]. Because this is an infinite process, in practice one has to abort the iteration at some point which is caused by a *termination condition* (line 28). In [Finthammer, 2012], two possible termination conditions are proposed:

1. Stop after the k-th iteration if $|1 - \eta_{(k),i}| < \delta$ for $i = 1, \ldots, n$ where δ is an appropriate threshold. In this case, the scaling factor is near to 1 and scaling the probability distribution $\mathcal{P}_k$ further has little effect.
2. Stop if $|\mathcal{P}_k(r_i) - p_i| < \delta$ for $i = 1, \ldots, n$. Hereby $\mathcal{P}_k(r_i)$ is the probability of the i-th conditional in $\mathcal{B}_{\mathcal{R}}$ with respect to the current probability distribution $\mathcal{P}_k$ which indeed should equal the input probability p_i of the conditional r_i best possible to ensure that the approximation (nearly) models $\mathcal{R}$.

Testing the second termination condition is computational expensive because in each iteration step the probabilities $\mathcal{P}_k(r_i)$ for $i = 1, \ldots, n$ have to be calculated. A third, more straightforward termination condition is to stop the iteration after the k-th iteration step if $|\mathcal{P}_k(\omega) - \mathcal{P}_{k-1}(\omega)| < \delta$ for $\omega \in \Omega(\Sigma)$ holds which would be a very generic termination condition for such kind of approximations. However, this termination condition is inappropriate because of the $|\Omega(\Sigma)|$-many checks.

The GIS-algorithm as proposed in [Finthammer, 2012] has been implemented as a plugin for the KReator system [Beierle et al., 2010]. KReator is an integrated development environment for relational probabilistic knowledge representation, reasoning, and learning.

The improvements on the GIS-algorithm made in [Finthammer and Beierle, 2014] focus on the runtime problem of the algorithm caused by the iterations over possible worlds and, hence, are in line with our research. Possible worlds are aggregated to equivalence classes according to their conditional structure (cf. Definition 3.2.6) such that the improved GIS-algorithm iterates over these equivalence classes instead of the possible worlds. As the cardinality of the equivalence classes has to be taken into account, too, the term *weighted conditional impact* (WCI) is introduced in [Finthammer and Beierle, 2014] which denotes a tuple that assigns to each equivalence class the conditional structure of the possible worlds within this equivalence class as well as the cardinality of the respective equivalence class. The authors of [Finthammer and Beierle, 2014] have shown that it is possible to overcome the iterations over possible worlds which have to be performed in the original GIS-algorithm by considering WCIs instead. This constitutes an important achievement towards domain-lifted calculations because the number of the equivalence classes with respect to the conditional structures of the possible worlds and, therewith, the number of the different WCIs is bounded polynomially in the domain size k as we show next.

Proposition 5.2.1: Upper Bound for Number of Equivalence Classes
Let Σ be a finite signature, and let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a consistent knowledge base from $\mathcal{RPCL}(\Sigma)$. Further, let e be the number of the equivalence classes induced by the equivalence relation $\sim_{\mathcal{B}_{\mathcal{R}}}$ (cf. Definition 3.2.6). Then, e is bounded polynomially in k because of the inequality

$$e \leq \prod_{r \in \mathcal{B}_{\mathcal{R}}} \frac{1}{2} \cdot \left(k^{2 \cdot \mathsf{FreeVar}(r)} + 3 \cdot k^{\mathsf{FreeVar}(r)} + 2\right).$$

Proof. Let $n = |\mathcal{B}_{\mathcal{R}}|$. First, we consider the case $n = 1$. The conditional structure of a possible world $\omega \in \Omega(\Sigma)$ solely depends on how many instances of the only conditional $r \in \mathcal{B}_{\mathcal{R}}$ are verified respectively falsified in ω then. The number of instances of r is $|\mathsf{Inst}_{\Sigma}(r)| = k^{\mathsf{FreeVar}(r)}$. Hence, at most $k^{\mathsf{FreeVar}(r)}$-many instances of r can be verified in ω. The remaining instances are either falsified or not applicable, which leads to the following maximal number of combinations of verifications and falsifications (here, we abbreviate $K = k^{\mathsf{FreeVar}(r)}$):

$$\begin{aligned}
&\sum_{l=0}^{K} \sum_{m=0}^{K-l} 1 \\
&= \sum_{l=0}^{K} (K - l + 1) \\
&= \sum_{l=0}^{K} K - \sum_{l=0}^{K} l + \sum_{l=0}^{K} 1 \\
&= (K+1) \cdot K - \frac{K \cdot (K+1)}{2} + (K+1) \\
&= (K+1) \cdot (K - \frac{K}{2} + 1) \\
&= \frac{1}{2} \cdot K^2 + \frac{3}{2} \cdot K + 1 \\
&= \frac{1}{2} \cdot (K^2 + 3 \cdot K + 2).
\end{aligned}$$

Note that in general not all of these combinations, i.e., different conditional structures, are realizable because of the logical constraints represented by the conditional, but $\frac{1}{2} \cdot (K^2 + 3 \cdot K + 2)$ definitely constitutes an upper bound for the number of different conditional structures.

Now let $n \geq 1$ be an arbitrary natural number. In order to obtain an upper bound for e in this case, we simply take the product of the value determined above over all conditionals. Then,

$$e \leq \prod_{r \in \mathcal{B}_{\mathcal{R}}} \frac{1}{2} \cdot \left(k^{2 \cdot \mathsf{FreeVar}(r)} + 3 \cdot k^{\mathsf{FreeVar}(r)} + 2 \right).$$

Note that if we build the equivalence classes over the potentially positive possible worlds from $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ instead of $\Omega(\Sigma)$, then the number of equivalence classes can only decrease. ☐

Despite the use of WCIs, the improved GIS-algorithm as proposed in [Finthammer and Beierle, 2014] is not suitable for domain-lifted calculations because it requires a precalculation of the weighted conditional impacts by iterating over all possible worlds once. Hence, the problem of getting rid of the iterations over possible worlds is only shifted to this preprocessing step—which of course reduces computational costs because the approximation with the improved GIS-algorithm can be performed much faster afterwards.

▶ Condensed Iterative Scaling

Like the improved GIS-algorithm from [Finthammer and Beierle, 2014], *condensed iterative scaling* (CIS) is an advancement of the original generalized iterative scaling algorithm GIS (cf. Figure 5.1) which stands out due to the fact that it is free of any iterations over possible worlds. In contrast to the improved GIS-algorithm, CIS makes use of an even more compact representation of conditional structures than weighted conditional impacts, the input of the improved GIS-algorithm. Actually, the additional, significant value of CIS originates from the fact that CIS exploits conditional structures of sets of possible worlds in parallel, based on Definition 3.2.3.

The basic idea of condensed iterative scaling is to approximate the ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$ of a Σ-p-consistent knowledge base $\mathcal{R}$ instead of the maximum entropy distribution $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$, like it is done in *improved iterative scaling*. Therewith, loops in CIS iterate over the non-factual conditionals in $\mathcal{B}_{\mathcal{R}}$ instead of the possible worlds in $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$. The critical point that remains after this transformation is the need to evaluate a sum over possible worlds in order to set up the scaling factors for adjusting the approximation of $\vec{\alpha}_{\Sigma,\mathcal{R}}$, similar to the sum in the scaling factors of the original GIS-algorithm (cf. Figure 5.1, line 23). This evaluation, however, is outsourced from the CIS-algorithm and moved to an external oracle so that calculations on the basis of possible worlds are totally removed from CIS.

Input: Σ-p-consistent knowledge base $\mathcal{R} = (\mathcal{F}_\mathcal{R}, \mathcal{B}_\mathcal{R})$ with $\mathcal{B}_\mathcal{R} = \{r_1, \ldots, r_n\}$ and $r_i = (\psi_i|\phi_i)_{\mathsf{CS}_i}[p_i]$ for $i = 1, \ldots, n$, precision $\epsilon > 0$
Output: Approximation $\vec{\alpha}^*_{\Sigma,\mathcal{R}}$ of Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$, normalization constant α^*_0

```
# Initialize with uniform alpha-values
FOR $i = 1, \ldots, n$:
  $\alpha_{(0),i} = 1$

# Precalculate constants
$G = \sum_{i=1}^{n} |\mathsf{Inst}_\Sigma(r_i)|$

# Initialize iteration counter
$k = 0$

# Iteratively adjust alpha-values
REPEAT
  $k = k + 1$
  FOR $i = 1, \ldots, n$:
    $\hat{\eta}_{(k),i} = \left(1 + \frac{\tau^{r_i}_{\Sigma,\mathcal{R}}(\alpha_{(k-1),1},\ldots,\alpha_{(k-1),n},\alpha^{-p_1}_{(k-1),1},\ldots,\alpha^{-p_n}_{(k-1),n},-p_i)}{p_i \cdot |\mathsf{Inst}_\Sigma(r_i)| \cdot \tau_{\Sigma,\mathcal{R}}(\alpha_{(k-1),1},\ldots,\alpha_{(k-1),n},\alpha^{-p_1}_{(k-1),1},\ldots,\alpha^{-p_n}_{(k-1),n})}\right)^{-1/G}$
                                          # Call oracle and calculate scaling factors
    $\alpha_{(k),i} = \alpha_{(k-1),i} \cdot \hat{\eta}_{(k),i}$                # Scale alpha-values
UNTIL $\max\{|\alpha_{(k),i} - \alpha_{(k-1),i}| < \epsilon \mid i = 1, \ldots, n\}$

# Return approximation of Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$
$\alpha_{(k),0} = \tau_{\Sigma,\mathcal{R}}(\alpha_{(k),1}, \ldots, \alpha_{(k),n}, \alpha^{-p_1}_{(k),1}, \ldots, \alpha^{-p_n}_{(k),n})^{-1}$
RETURN $\vec{\alpha}^*_{\Sigma,\mathcal{R}} = (\alpha^*_1, \ldots, \alpha^*_n) = (\alpha_{(k),1}, \ldots, \alpha_{(k),n})$
                                          (and normalization constant $\alpha^*_0 = \alpha_{(k),0}$)
```

Algorithm 5.2: Algorithm CIS which computes an approximation $\vec{\alpha}^*_{\Sigma,\mathcal{R}}$ of the Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$.

The CIS-algorithm is presented in Figure 5.2. It takes a Σ-p-consistent knowledge base $\mathcal{R} = (\mathcal{F}_\mathcal{R}, \mathcal{B}_\mathcal{R})$ with $\mathcal{B}_\mathcal{R} = \{r_1, \ldots, r_n\}$ as well as a precision $\varepsilon > 0$ as an input and calculates an approximation $\vec{\alpha}^*_{\Sigma,\mathcal{R}}$ of the Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$. In lines 1-3 the approximations of the alpha-values, the components of $\vec{\alpha}_{\Sigma,\mathcal{R}}$, are initialized by 1. Afterwards, they are iteratively adjusted (lines 11-18) by a multiplication with the scaling factor (line 15)

$$\hat{\eta}_{(k),i} = \left(1 + \frac{\tau^{r_i}_{\Sigma,\mathcal{R}}(\alpha_{(k-1),1}, \ldots, \alpha_{(k-1),n}, \alpha^{-p_1}_{(k-1),1}, \ldots, \alpha^{-p_n}_{(k-1),n}, -p_i)}{p_i \cdot |\mathsf{Inst}_\Sigma(r_i)| \cdot \tau_{\Sigma,\mathcal{R}}(\alpha_{(k-1),1}, \ldots, \alpha_{(k-1),n}, \alpha^{-p_1}_{(k-1),1}, \ldots, \alpha^{-p_n}_{(k-1),n})}\right)^{-1/G},$$

where, as usual, k is the iteration counter, $i = 1, \ldots, n$ are the indices of the conditionals in $\mathcal{B}_\mathcal{R}$, $|\mathsf{Inst}_\Sigma(r_i)|$ is the number of the ground instantiations of the conditional r_i, and G is the sum of these numbers summed up over all conditionals in $\mathcal{B}_\mathcal{R}$ (line 6). The functions

$$\tau^{r_i}_{\Sigma,\mathcal{R}}(x_1, \ldots, x_n, y_1, \ldots, y_n, z) = \sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} (\mathsf{ver}_{\Sigma,r_i}(\omega) + z \cdot \mathsf{app}_{\Sigma,r_i}(\omega)) \cdot \prod_{j=1}^{n} x_j^{\mathsf{ver}_{\Sigma,r_j}(\omega)} \cdot y_j^{\mathsf{app}_{\Sigma,r_j}(\omega)} \tag{5.1}$$

and

$$\tau_{\Sigma,\mathcal{R}}(x_1, \ldots, x_n, y_1, \ldots, y_n) = \sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} \prod_{j=1}^{n} x_j^{\mathsf{ver}_{\Sigma,r_j}(\omega)} \cdot y_j^{\mathsf{app}_{\Sigma,r_j}(\omega)}, \tag{5.2}$$

that are invoked in $\hat{\eta}_{(k),i}$, need to be explained. They are polynomial reformulations of the left-hand sides of the maximum entropy equation system (4.19) (i.e., the left-hand sides of the characterization of the aggregating semantics for the conditionals in $\mathcal{B}_\mathcal{R}$ in which the maximum entropy model is plugged in) and the objective function of the dual maximum entropy optimization problem (4.12) (i.e., the left-hand side of the normalization condition for the maximum entropy model), respectively. That is, for $i = 1, \ldots, n$, the polynomial $\tau^{r_i}_{\Sigma,\mathcal{R}}(x_1, \ldots, x_n, y_1, \ldots, y_n, z)$ refers to the expression

$$\sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} f_{\Sigma,i}(\omega) \cdot \prod_{j=1}^{n} x_i^{f_{\Sigma,j}(\omega)},$$

and $\tau_{\Sigma,\mathcal{R}}(x_1, \ldots, x_n, y_1, \ldots, y_n)$ refers to

$$\sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} \prod_{i=1}^{n} x_i^{f_{\Sigma,i}(\omega)}.$$

Note that the prefactors and the exponents which occur in the polynomials $\tau^{r_i}_{\Sigma,\mathcal{R}}$ and $\tau_{\Sigma,\mathcal{R}}$ are the numbers of verifications and applicabilities of the conditionals in $\mathcal{B}_{\mathcal{R}}$ which can be precalculated and do not change during the iteration process. Everything else which occurs in (4.19) or (4.12) and which may be adapted during the iteration process is encapsulated in the variables $x_1, \ldots, x_n, y_1, \ldots, y_n$. The variable z is a placeholder for the negated probability of r_i which is not an integer number and, therefore, detached from the polynomial representation as well.

Both (4.19) and (4.12) can be recovered from $\tau^{r_i}_{\Sigma,\mathcal{R}}$ and $\tau_{\Sigma,\mathcal{R}}$, respectively, by plugging in α_i for x_i, $\alpha_i^{-p_i}$ for y_i, and $-p_i$ for z, each for $i = 1, \ldots, n$, and by rearranging expressions by exploiting the definition of the feature functions

$$f_{\Sigma,i}(\omega) = \mathsf{ver}_{\Sigma,i}(\omega) - p_i \cdot \mathsf{app}_{\Sigma,i}(\omega).$$

In this chapter, we assume that the polynomials $\tau^{r_i}_{\Sigma,\mathcal{R}}$ and $\tau_{\Sigma,\mathcal{R}}$ are set up and evaluated by an oracle as mentioned above. We will discuss how such an oracle could work in Section 6.4 when we derive these polynomials with typed model counting.

Now, we prove the correctness of the CIS algorithm by arguing that the sequence

$$\vec{\alpha}_{(k)} = (\alpha_{(k),1}, \ldots, \alpha_{(k),n})_{k \in \mathbb{N}_0}$$

defined in this algorithm (cf. Figure 5.2) converges towards the Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$. From the convergence of the sequence it directly follows that CIS is sound and terminates.

Proposition 5.2.2: Correctness of Condensed Iterative Scaling
Let Σ be a finite signature, and let $\mathcal{R}$ be a Σ-p-consistent knowledge base. Then, the sequence $\vec{\alpha}_{(k)} = (\alpha_{(k),1}, \ldots, \alpha_{(k),n})_{k \in \mathbb{N}_0}$ as defined in the algorithm CIS (cf. Figure 5.2) converges component-wise towards the Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$.

Proof. If one applies standard generalized iterative scaling to solve the maximum entropy problem (cf. the GIS-algorithm in Figure 5.1), one iteratively scales the probabilities of the possible worlds $\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ via (cf. lines 25 and 27 of GIS in Figure 5.1)

$$\mathcal{P}_k(\omega) = \tfrac{1}{\zeta} \cdot \mathcal{P}_{k-1}(\omega) \cdot \textstyle\prod_{j=0}^{n} (\eta_{(k),j})^{\tilde{f}_{\Sigma,j}(\omega)}, \tag{5.3}$$

where k is the iteration index,

$$\zeta = \sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}_{k-1}(\omega) \cdot \prod_{j=0}^{n} (\eta_{(k),j})^{\tilde{f}_{\Sigma,j}(\omega)}$$

is a normalizing constant, and where

$$\eta_{(k),i} = \mu_i \cdot \Big(\sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}} \mathcal{P}_{(k-1)}(\omega) \cdot \tilde{f}_{\Sigma,i}(\omega) \Big)^{-1}, \quad i = 0, \ldots, n, \tag{5.4}$$

with μ_i as defined in line 13 of GIS, are the scaling factors of this algorithm (cf. line 23 of GIS). With these scaling factors, one can iteratively derive the values $\hat{\alpha}_{(k),i} = \hat{\alpha}_{(k-1),i} \cdot \eta_{(k),i}$ for $i = 0, 1, \ldots, n$ when beginning with $\hat{\alpha}_{(0),i} = 1$. Therewith,

$$\alpha_{(k),i} = \left(\frac{\hat{\alpha}_{(k),i}}{\hat{\alpha}_{(k),0}}\right)^{1/G}, \quad i = 1, \ldots, n,$$

because $\hat{\alpha}_{(k),i}$ is $\alpha_{(k),i}$ up to normalization, such that

$$\alpha_{(k),i} = \alpha_{(k-1),i} \cdot \left(\frac{\eta_{(k),i}}{\eta_{(k),0}}\right)^{1/G}, \quad i = 1, \ldots, n. \tag{5.5}$$

As a direct consequence of both the convergence of GIS and the strong duality between the primal and the dual maximum entropy optimization problem, it follows that $(\alpha_{(k),0}, \alpha_{(k),1}, \ldots, \alpha_{(k),n})_{k \in \mathbb{N}_0}$ converges against $\vec{\alpha}_{\Sigma,\mathcal{R}}$. It remains to show that the iteration specification (5.5) can be reformulated to line 17 of CIS. The convergence of $(\alpha_{(k),0})_{k \in \mathbb{N}_0}$ to the normalization constant α_0 of the Σ-ME-vector is trivial then. For this, recursively plug in the predecessor $\mathcal{P}_{k-1}(\omega)$ of $\mathcal{P}_k(\omega)$ into equation (5.3) and use that $\hat{\alpha}_{(0),i} = 1$ for $i = 1, \ldots, n$ and $\mathcal{P}_0(\omega) = |\Omega_{\mathcal{F}_\mathcal{R}}|^{-1}$ for $\omega \in \Omega_{\mathcal{F}_\mathcal{R}}$ (GIS starts iterating from the uniform distribution) which leads to

$$\mathcal{P}_k(\omega) = \frac{1}{\zeta} \cdot \prod_{j=0}^{n} (\hat{\alpha}_{(k),j})^{\tilde{f}_{\Sigma,j}(\omega)}. \tag{5.6}$$

Now, we insert (5.6) into (5.4), which removes the probabilities $\mathcal{P}_k(\omega)$ from the iteration specification of $\hat{\alpha}_{(k),i}$, and get

$$\alpha_{(k),i} = \alpha_{(k-1),i} \cdot \left(\frac{\mu_0 \cdot \sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}} \tilde{f}_{\Sigma,i}(\omega) \cdot \prod_{j=0}^{n} (\hat{\alpha}_{(k),j})^{\tilde{f}_{\Sigma,j}(\omega)}}{\mu_i \cdot \sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}} \tilde{f}_{\Sigma,0}(\omega) \cdot \prod_{j=0}^{n} (\hat{\alpha}_{(k),j})^{\tilde{f}_{\Sigma,j}(\omega)}}\right)^{-1/G}.$$

Finally, we obtain the scaling rule in line 17 of CIS by plugging in μ_i and $\tilde{f}_{\Sigma,i}(\omega)$ for $i = 0, 1 \ldots, n$ into the last equation. □

From an approximation $\vec{\alpha}^*_{\Sigma,\mathcal{R}} = (\alpha^*_1, \ldots, \alpha^*_n)$ of the Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$, an approximation of the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ of $\mathcal{R}$ can be computed by

$$\mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}(\omega) = \begin{cases} \alpha^*_0 \cdot \prod_{i=1}^{n} (\alpha^*_i)^{f_{\Sigma,i}(\omega)} & \text{if} \quad \omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma) \\ 0 & \text{otherwise} \end{cases}. \tag{5.7}$$

Indeed, $\mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}$ is a probability distribution as the following proposition shows.

Proposition 5.2.3: Approximation of Maximum Entropy Model
Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a Σ-p-consistent knowledge base from $\mathcal{RPCL}(\Sigma)$, and let $\vec{\beta} \in \mathbb{R}^n_{>0}$ be a vector of positive real numbers. Then,

$$\mathcal{P}_{\vec{\beta},\mathcal{R}}(\omega) = \begin{cases} \dfrac{\prod_{i=1}^{n} (\beta_i)^{f_{\Sigma,i}(\omega)}}{\sum_{\omega' \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \prod_{i=1}^{n} (\beta_i)^{f_{\Sigma,i}(\omega')}}, & \text{if} \quad \omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma) \\ 0, & \text{otherwise} \end{cases} \tag{5.8}$$

is a probability distribution over $\Omega(\Sigma)$. In particular, this holds for $\mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}(\omega)$ defined in (5.7).

Proof. Because β_i is positive, the expression $(\beta_i)^{f_{\Sigma,i}(\omega)}$ is well-defined and positive. As a consequence, $\prod_{i=1}^{n} (\beta_i)^{f_{\Sigma,i}(\omega)}$ is positive, too. The division through $\sum_{\omega' \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \prod_{i=1}^{n} (\beta_i)^{f_{\Sigma,i}(\omega')}$ in (5.8) means a normalization and, hence, $\mathcal{P}_{\vec{\beta},\mathcal{R}}(\omega) \in (0,1)$. Further, one has

$$\sum_{\omega \in \Omega(\Sigma)} \mathcal{P}_{\vec{\beta},\mathcal{R}}(\omega) = \sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \mathcal{P}_{\vec{\beta},\mathcal{R}}(\omega) = \frac{\sum_{\omega' \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \prod_{i=1}^{n} (\beta_i)^{f_{\Sigma,i}(\omega')}}{\sum_{\omega' \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \prod_{i=1}^{n} (\beta_i)^{f_{\Sigma,i}(\omega')}} = 1.$$

It remains to show that $\mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}$ is of the form (5.8). This, however, is clear because (cf. line 21 of CIS)

$$\begin{aligned} \alpha_0^* &= \tau_{\Sigma,\mathcal{R}}(\alpha_1^*, \ldots, \alpha_n^*, (\alpha_1^*)^{-p_1}, \ldots, (\alpha_n^*)^{-p_n}) \\ &= \sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \prod_{j=1}^{n} (\alpha_1^*)^{\mathsf{ver}_{\Sigma,r_j}(\omega)} \cdot ((\alpha_j^*)^{-p_j})^{\mathsf{app}_{\Sigma,r_j}(\omega)} \\ &= \sum_{\omega \in \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)} \prod_{j=1}^{n} (\alpha_j^*)^{f_{\Sigma,j}(\omega)}. \end{aligned}$$

Note that $\alpha_i^* > 0$ holds for $i = 1, \ldots, n$ because in the CIS-algorithm $\alpha_{(0),i}$ is initialized with 1 (cf. Figure 5.2, line 3) and multiplied with a positive scaling factor $\eta_{(k),i} > 0$ in every step k with $k > 0$ (line 17). The fact that $\eta_{(k),i} > 0$ holds can be proven as follows: By definition of $\tau^{r_i}_{\Sigma,\mathcal{R}}$ (5.1) and $\tau_{\Sigma,\mathcal{R}}$ (5.2),

$$\tau^{r_i}_{\Sigma,\mathcal{R}}(x_1, \ldots, x_n, y_1, \ldots, y_n, -p_i) > -p_i \cdot |\mathsf{Inst}_\Sigma(r_i)| \cdot \tau_{\Sigma,\mathcal{R}}(x_1, \ldots, x_n, y_1, \ldots, y_n)$$

for all $x_1, \ldots, x_n, y_1, \ldots, y_n \in \mathbb{R}_{\geq 0}$ and $p_i \in (0,1)$. The left-hand side of this inequation is *strictly* greater than the right-hand side because there is at least one possible world in which r_i is verified at least once. Otherwise, $p_i > 0$ would not be possible because the numerator of the fraction in the definition of the aggregating semantics would be 0 (resp. $\mathcal{R}$ would not be Σ-consistent). Therewith,

$$\frac{\tau^{r_i}_{\Sigma,\mathcal{R}}(\alpha_{(k-1),1},\ldots,\alpha_{(k-1),n},\alpha^{-p_1}_{(k-1),1},\ldots,\alpha^{-p_n}_{(k-1),n},-p_i)}{p_i\cdot|\mathsf{Inst}_\Sigma(r_i)|\cdot\tau_{\Sigma,\mathcal{R}}(\alpha_{(k-1),1},\ldots,\alpha_{(k-1),n},\alpha^{-p_1}_{(k-1),1},\ldots,\alpha^{-p_n}_{(k-1),n})}>-1$$

holds from which $\eta_{(k),i} > 0$ follows. $\square$

At this point, it is a fair objection to say that the critical computations with exponential dependency on the domain size k are just shifted from the iterative scaling algorithm CIS to the oracle and, in the end, nothing is gained. However, in the following chapter we will discuss how to compute the polynomials $\tau_{\Sigma,\mathcal{R}}$ and $\tau^{r_i}_{\Sigma,\mathcal{R}}$ systematically—and in some cases tractably—by the use of the concept of *typed model counting*. It remains an open question how the error in approximating the ME-vector propagates into errors in the calculation of the maximum entropy probabilities.

6

First-Order Typed Model Counting

In this chapter, we present the concept of *typed model counting.* Typed model counting lifts first-order typed model counting techniques to the conditional setting based on a compilation of knowledge bases into so-called *structured sentences.* Structured sentences reflect the three-valued evaluation of the ground instances of conditionals by the use of abstract *structure elements* which are directly incorporated in the formulas. We exploit typed model counting to compute the input of the condensed iterative scaling algorithm from Section 5.2. First, we give a general introduction to *first-order model counting* and its variants, in particular *algebraic model counting* (Section 6.1), before we discuss typed model counting in detail (Section 6.2). In Section 6.3, we discuss the connection between typed model counting and algebraic model counting. Eventually, in Section 6.4, we show how typed model counting can be utilized for conditional knowledge compilation, in particular in the context of maximum entropy reasoning.

6.1 First-Order Model Counting

First-order model counting is the task of counting the models of relational sentences (cf., e.g., Van den Broeck [2016]). While this seems to be a purely theoretical task at first sight, it becomes very useful in probabilistic applications, for instance, because model counting techniques can be used to compute probabilities of complex sentences by weighting the model counts with the elementary probabilities assigned to the models. Roughly speaking, the more efficient the model counting task is performed, the more efficient the probabilities can be computed.

In this section, we give an introduction to first-order model counting in general, and then discuss the *weighted first-order model counting task* in more detail. In particular, we point out that the weights which are assigned to models can be elements of very general algebraic structures and are not restricted to probabilities. This is a potential that one makes use of in *algebraic model counting.* Afterwards, we discuss normal forms of relational sentences which are especially suited for the model counting task. Finally, we show that every sentence in $\mathcal{RL}^S(\Sigma)$ can be compiled into such a normal form.

▶ Introduction To First-Order Model Counting

Given a finite signature Σ, we denote the *first-order model count* of a sentence $\phi \in \mathcal{RL}^S(\Sigma)$, i.e., the number of its Σ-models, by

$$\mathsf{FOMC}_\Sigma(\phi) = |\mathsf{Mod}_\Sigma(\phi)|. \tag{6.1}$$

The *first-order model counting task* is to compute $\mathsf{FOMC}_\Sigma(\phi)$ then. By testing for each interpretation $I \in \mathsf{Int}_\Sigma(\Sigma)$ whether $I \in \mathsf{Mod}_\Sigma(\phi)$ holds or not, this task can be performed in time exponential in the domain size $k = |\mathsf{Const}_\Sigma|$ in a naïve way. The essence of *domain-lifted* first-order model counting is to compute $\mathsf{FOMC}_\Sigma(\phi)$ in time polynomial in k (or even better) instead, which is sometimes possible by compiling the sentence ϕ into a suitable normal form first, before the actual counting takes place. Obviously, if $\phi \equiv_\Sigma \phi'$, then $\mathsf{FOMC}_\Sigma(\phi) = \mathsf{FOMC}_\Sigma(\phi')$ holds, which justifies the precompilation of sentences into normal forms.

In the context of domain-lifted model counting, the essence of normal forms is to exploit symmetries and independencies of constants within the sentence, which are particularly caused by “indistinguishable” domain elements as well as further combinatorial arguments, to speed up counting. The most common examples of indistinguishable domain elements are the “unnamed” domain elements which are represented by constants that do not explicitly occur in the sentence. In this case, the sentence itself “gives no reason” to distinguish between these unnamed elements, which is reflected by a factorization of the model count. On a technical level, the combinatorial arguments come into view by specific properties of the connectives $\neg$, $\wedge$, $\vee$, $\exists$, and $\forall$ which do not hold in general but in proper normal forms. To convey an idea of these combinatorial counting techniques, we give a simple yet illustrative example which can be understood without formal definitions of the combinatorial arguments.

Example 6.1.1

We ask for the Σ-model count of the sentence

$$\phi = \forall X.(\mathsf{male}(X) \Rightarrow \neg\mathsf{female}(X)).$$

> “For all individuals it holds that if they are male, then they are not female.”

Hereby, Const_Σ shall be an arbitrary but finite set and

$$\mathsf{Pred}_\Sigma = \{\mathsf{female}/1, \mathsf{male}/1\}.$$

Because for all interpretations $I \in \mathsf{Int}(\Sigma)$ and for all constants $c \in \mathsf{Const}_\Sigma$, the evaluation of $I(\mathsf{male}(c) \Rightarrow \neg\mathsf{female}(c))$ is independent of the evaluation of $I(\mathsf{male}(c') \Rightarrow \neg\mathsf{female}(c'))$ where c' is any different constant from Const_Σ, i.e., $c' \neq c$, one can count the models of the "subsentence"

$$\mathsf{male}(c) \Rightarrow \neg\mathsf{female}(c) \tag{6.2}$$

"If c is male, then c is not female."

of ϕ (more precisely, of the instances of the universal quantification in ϕ) with respect to its restricted signature $\Sigma' = (\{c\}, \mathsf{Pred}_\Sigma)$ and exponentiate this count with the number of constants in Const_Σ in order to observe the total model count of ϕ. The sentence (6.2) is satisfied if either c is not male or c is male but not female. This can be directly read from following sentence which is equivalent to (6.2):

$$(\mathsf{male}(c) \wedge \neg\mathsf{female}(c)) \vee (\neg\mathsf{male}(c) \wedge (\mathsf{female}(c) \vee \neg\mathsf{female}(c)).$$

In case of $\neg\mathsf{male}(c)$, the evaluation of $\mathsf{female}(c)$ is irrelevant for the satisfaction of the sentence ϕ. We mention the tautology $(\mathsf{female}(c) \vee \neg\mathsf{female}(c))$ in this case anyway in order to indicate that there are two different subcases in which $\neg\mathsf{male}(c)$ holds—either $\mathsf{female}(c)$ or $\neg\mathsf{female}(c)$ holds in addition—which is relevant for counting the models of ϕ. Eventually, we rearrange ϕ to the sentence

$$\phi' = \bigwedge_{c \in \mathsf{Const}_\Sigma} \big((\mathsf{male}(c) \wedge \neg\mathsf{female}(c)) \vee (\neg\mathsf{male}(c) \wedge (\mathsf{female}(c) \vee \neg\mathsf{female}(c)))\big)$$

that is equivalent to ϕ and from which we can easily read its model count—and therewith the model count of ϕ—to

$$\mathsf{FOMC}_\Sigma(\phi') = \mathsf{FOMC}_\Sigma(\phi) = (1+2)^{|\mathsf{Const}_\Sigma|}. \tag{6.3}$$

In Example 6.1.3 we will show that ϕ' is the translation of ϕ into the normal form called *smooth deterministic decomposable negation normal form* (sd-DNNF) that is especially suited for the model counting task. The summand 1 in (6.3) refers to the case in which $\mathsf{male}(c)$ and $\neg\mathsf{female}(c)$ hold while the summand 2 is the number of cases in which $\neg\mathsf{male}(c)$ holds. Here, it is important that the disjuncts $\mathsf{male}(c) \wedge \neg\mathsf{female}(c)$ and $\neg\mathsf{male}(c) \wedge (\mathsf{female}(c) \vee \neg\mathsf{female}(c))$ are mutually exclusive so that cases are not counted twice.

Obviously, model counting is restricted to the case of finite domains, and the model count of a sentence usually depends on the domain size $k = |\mathsf{Const}_\Sigma|$. Please see [Van den Broeck, 2016] for a more in-depth introduction to first-order model counting and recommendations for further reading.

▶ Weighted First-Order Model Counting

The task of first-order model counting is of high importance in relational probabilistic reasoning because the probability of a sentence $\phi \in \mathcal{RL}^S(\Sigma)$ is the sum of the probabilities of its models, here denoted as possible worlds. That is, we have

$$\mathcal{P}(\phi) = \sum_{\omega \in \Omega(\Sigma)\colon \omega \models_\Sigma \phi} \mathcal{P}(\omega),$$

where $\mathcal{P}$ is a probability distribution over $\Omega(\Sigma)$. Thus, by counting the models/possible worlds of the sentence ϕ while weighting them with their probability $\mathcal{P}(\omega)$ it is possible to infer the probability $\mathcal{P}(\phi)$ of the sentence ϕ from the elementary probabilities of the probability distribution $\mathcal{P}$. More generally, the variation of first-order model counting which is called *weighted first-order model counting* [Chavira and Darwiche, 2008] means the task of counting the *weighted models* of a sentence $\phi \in \mathcal{RL}^S(\Sigma)$,

$$\mathsf{WFOMC}_\Sigma(\phi) = \sum_{\omega \in \Omega(\Sigma)\colon \omega \models \phi} W(\omega),$$

where $W(\omega) \in \mathbb{R}$ is any real valued weight assigned to the possible world $\omega \in \Omega(\Sigma)$. Considering probabilities as weights is the major instance of the weighted model counting task.

A common assumption in weighted first-order model counting is that the weights are originally assigned not to the models but to the ground literals in $\mathsf{grLit}(\Sigma)$ (cf. [Van den Broeck, 2016]) and that the weights of the models can then be computed to $W(\omega) = \prod_{l \in \mathsf{grLit}(\Sigma)\colon \omega \models l} W(l)$. Therewith, the weighted model counting task becomes

$$\mathsf{WFOMC}'_\Sigma(\phi) = \sum_{\omega \models \phi} \prod_{l \in \mathsf{grLit}(\Sigma)\colon \omega \models l} W(l). \tag{6.4}$$

In the case of probabilistic weights, i.e., if $W(l) = p$ for a probability value p, we have that $W(l) = p$ implies $W(\neg l) = 1 - p$ because the probabilities of an event and its complement must sum up to 1. In general the weights of literals and their negative counterparts can be chosen independently in weighted first-order model counting, though.

The assumption that the weights are assigned to the ground literals is essential for the efficiency of many counting techniques because it reduces the number of weights from exponentially many to polynomially many if measured in $k = |\mathsf{Const}_\Sigma|$. Further, in (6.4) the weight of a ground literal can be separated from the sum as a prefactor if the ground literal occurs in every model of ϕ, and factorizing is *the* essential tool of efficient model counting (often visualized as $\sum\prod \to \prod\sum$; the commuting of building sums and products). In the context of probabilistic reasoning, the assumption that the probabilities of possible worlds are the product of the probabilities of the ground literals, however, means that the ground atoms are statistically independent which is a very strong assumption that we do not want to make in this thesis.

An alternative to assigning weights to literals instead of possible worlds in order to overcome the problem of exponentially many (different) weights is to force weights to be identical. In *symmetric weighted first-order model counting* [Beame et al., 2015] both of these assumptions are combined and weights of ground literals $l', l'' \in \mathsf{grLit}(\Sigma)$ are assumed to be the same, i.e., $W(l') = W(l'')$, if l' and l'' are groundings of the same literal from $\mathsf{Lit}(\Sigma)$. The effect of this assumption is that products over ground literals which refer to the same literal can be aggregated to powers, and sums to a single expression with a prefactor which equals the number of proper groundings. In addition, less weights have to be stored.

Ordinary first-order model counting can be seen as an instance of weighted first-order model counting by assigning the weight $W(l) = 1$ to every ground literal $l \in \mathsf{grLit}(\Sigma)$. Note that in this case it does not mean a restriction to the model counting task when the weights are assigned to the ground literals because it holds that $\omega \in \mathsf{Mod}_\Sigma(\phi)$ if and only if $\prod_{l\in\mathsf{grLit}(\Sigma)\colon \omega\models l} 1 = 1$. Hence,

$$\mathsf{FOMC}_\Sigma(\phi) = \sum_{\omega\in\Omega(\Sigma)\colon \omega\models\phi} \; \prod_{l\in\mathsf{grLit}(\Sigma)\colon \omega\models l} 1 \tag{6.5}$$

is an equivalent reformulation of the first-order model counting task (6.1). Also, the additional assumption of symmetric weights is trivially satisfied because the weights are all equal, namely 1. We have exploited this fact in Example 6.1.1 already.

▶ Algebraic First-Order Model Counting

Algebraic first-order model counting [Kimmig et al., 2017] generalizes weighted first-order model counting to the case where the weights assigned to the ground literals are elements from an arbitrary *Abelian semiring* $(\mathcal{S}, \oplus, \otimes, e^{\oplus}, e^{\otimes})$ (cf. Definition A.1.4 of Abelian semirings in the Appendix). While in Example 6.1.1 we have replaced every conjunction by the ordinary multiplication of numbers $\cdot$ and every disjunction by $+$ in order to count models, in algebraic model counting conjunctions are replaced by the multiplication $\otimes$ from $\mathcal{S}$ and disjunctions by $\oplus$. Because the algebraic structure $(\{\mathcal{P}(l) \mid l \in \mathsf{grLit}(\Sigma)\}, +, \cdot, 0, 1)$ where $\mathcal{P}$ is a joint probability distribution over the ground literals in $\mathsf{grLit}(\Sigma)$ indeed constitutes an Abelian semiring, algebraic first-order model counting subsumes the task of calculating $\mathcal{P}(\phi) = \sum_{\omega \in \Omega(\Sigma)\colon \omega \models \phi} \mathcal{P}(\omega)$ in the same way as weighted first-order model counting. However, algebraic first-order model counting is not limited to probabilities. Formally, the algebraic model count of a sentence $\phi \in \mathcal{RL}^S(\Sigma)$ is

$$\mathsf{AFOMC}^{\rho}_{\Sigma}(\phi) = \bigoplus_{\omega \in \Omega(\Sigma)\colon \omega \models \phi} \ \bigotimes_{l \in \mathsf{grLit}(\Sigma)\colon \omega \models l} \rho(l), \tag{6.6}$$

where $\rho : \mathsf{grLit}(\Sigma) \to \mathcal{S}$ maps the ground literals from $\mathsf{grLit}(\Sigma)$ to the elements of the semiring $\mathcal{S}$.

Note that algebraic model counting makes use of the same independence assumptions as weighted first-order model counting, namely that the algebraic elements from $\mathcal{S}$ are assigned to the ground literals in the first instance. From this assignment, we can deduce an assignment to interpretations $I \in \mathsf{Int}(\Sigma)$ via

$$I_{\rho} = \bigotimes_{l \in \mathsf{grLit}(\Sigma)\colon I(l)=1} \rho(l)$$

and to possible worlds $\omega \in \Omega(\Sigma)$ via

$$\omega_{\rho} = \bigotimes_{l \in \mathsf{grLit}(\Sigma)\colon \omega \models l} \rho(l).$$

For $I \in \mathsf{Int}(\Sigma)$, we call I_{ρ} *algebraic interpretation* (wrt. ρ) and define the algebraic interpretation of a general sentence $\phi \in \mathcal{RL}^S(\Sigma)$ wrt. ρ by $I_{\rho}(\phi) = I_{\rho}$ if $I \models_{\Sigma} \phi$, and by $I_{\rho}(\phi) = 0$ otherwise.

▶ Normal Form sd-DNNF

In the following, we discuss the *smooth deterministic decomposable negation normal form* [Darwiche, 2001a,b], sd-DNNF for short, which is a normal form of relational sentences that is especially suited for the first-order model counting task. This means that the model count of a sentence in sd-DNNF can be calculated in time polynomial in the length of the sentence in binary notation [Darwiche and Marquis, 2002]. For an overview of further normal forms and their appropriateness for different reasoning tasks, please see [Darwiche and Marquis, 2002] which provides a concise while thorough summary of the foundations of knowledge compilation.

For the sake of simplicity, in the following definition of sentences in sd-DNNF-normal form, we assume that the considered sentence is free of quantifications. This can be achieved by compiling every quantification into either a conjunction with domain constraints (cf. Section 2.3) in the case of universal quantification, or into a disjunction with domain constraints in the case of existential quantification beforehand. Recall that

$$\forall_{\mathsf{CS}} X.\phi \equiv \bigwedge_{c \in \mathsf{Sol}(\mathsf{CS})} \phi\langle X/c \rangle \quad \text{and} \quad \exists_{\mathsf{CS}} X.\phi \equiv \bigvee_{c \in \mathsf{Sol}(\mathsf{CS})} \phi\langle X/c \rangle \tag{6.7}$$

hold. Thus, these compilations lead to an equivalent reformulation of sentences while maintaining their compact representation, and the restriction to quantifier-free sentences does not mean a loss of expressivity. The gain is that we do not need to consider unhandy instantiations of variables in the definition of sentences in sd-DNNF since every quantifier-free sentence is free of variables.

Definition 6.1.2: sd-DNNF-Normal Form *(cf. [Darwiche, 2001a,b])*

Let Σ be a finite signature, and let $\phi \in \mathcal{RL}^S(\Sigma)$ be a quantifier-free sentence. Then, ϕ is in *negation normal form* (NNF) if negations in ϕ occur directly in front of ground atoms only. In addition, ϕ is *decomposable*, *deterministic*, and *smooth*, respectively, if the following conditions hold:

- The sentence ϕ is *decomposable* iff every conjunction $\phi_1 \wedge \ldots \wedge \phi_m$ in ϕ is *decomposable* which means that $\mathsf{grAtom}(\phi_i) \cap \mathsf{grAtom}(\phi_j) = \emptyset$ holds for all $i, j \in \{1, \ldots, m\}$ with $i \neq j$.
- The sentence ϕ is *deterministic* iff every disjunction $\phi_1 \vee \ldots \vee \phi_m$ in ϕ is *deterministic* which means that every two disjuncts out of this disjunction are *contradictory* (or *mutually exclusive*), where two disjuncts ϕ_i, ϕ_j with $i, j \in \{1, \ldots, m\}$ and $i \neq j$ are *contradictory* if $\phi_i \wedge \phi_j \equiv \bot$.

- The sentence ϕ is *smooth* iff both (1) every disjunction $\phi_1 \vee \ldots \vee \phi_m$ in ϕ is *smooth*, which means that $\mathsf{grAtom}(\phi_i) = \mathsf{grAtom}(\phi_j)$ holds for $i, j \in \{1, \ldots, m\}$, and (2) $\mathsf{grAtom}(\phi) = \mathsf{grAtom}(\Sigma)$ holds. The latter condition guarantees that all ground atoms from the underlying language $\mathcal{RL}(\Sigma)$ are mentioned in ϕ. This condition can be dropped if the background language which is considered is the language that is induced by the sentence ϕ.

With $\mathsf{sd\text{-}DNNF}(\Sigma) \subseteq \mathcal{RL}^S(\Sigma)$ we denote the set of all sentences $\phi \in \mathcal{RL}^S(\Sigma)$ which are in sd-DNNF-normal form, i.e., ϕ is in negation normal form as well as decomposable, deterministic, and smooth.

In order to indicate that a conjunction is decomposable we write $\wedge^d$ instead of $\wedge$. Analogously, deterministic and smooth disjunctions are indicated by $\vee^d$ and $\vee^s$, respectively. If a disjunction is both deterministic and smooth, then we write $\vee^{sd}$.

Example 6.1.3

The sentence

$$\phi' = \bigwedge_{c \in \mathsf{Const}_\Sigma} \big((\mathsf{male}(c) \wedge \neg\mathsf{female}(c)) \vee (\neg\mathsf{male}(c) \wedge (\mathsf{female}(c) \vee \neg\mathsf{female}(c)))\big)$$

from Example 6.1.1 is in sd-DNNF normal form. Obviously, ϕ' is in negation normal form because negations occur directly in front of $\mathsf{female}(c)$ or $\mathsf{male}(c)$ only. The outer conjunction with the domain constraint $c \in \mathsf{Const}_\Sigma$ is decomposable because the single conjuncts mention ground atoms with respect to different constants. More precisely, ϕ' is of the form $\phi' = \bigwedge_{c \in \mathsf{Const}_\Sigma} \phi_c$ with

$$\phi_c = (\mathsf{male}(c) \wedge \neg\mathsf{female}(c)) \vee (\neg\mathsf{male}(c) \wedge (\mathsf{female}(c) \vee \neg\mathsf{female}(c)))$$

and $\mathsf{grAtom}(\phi_c) = \{\mathsf{male}(c), \mathsf{female}(c)\}$ for $c \in \mathsf{Const}_\Sigma$, such that $\mathsf{grAtom}(\phi_c) \cap \mathsf{grAtom}(\phi_{c'}) = \emptyset$ holds if $c \neq c'$. Likewise, the two inner conjunctions $\mathsf{male}(c) \wedge \neg\mathsf{female}(c)$ and $\neg\mathsf{male}(c) \wedge (\mathsf{female}(c) \vee \neg\mathsf{female}(c))$ are decomposable because the single conjuncts mention either the ground atom $\mathsf{male}(c)$ or $\mathsf{female}(c)$.

In addition, the outer disjunction in ϕ_c is deterministic because the first disjunct only holds if $\mathsf{male}(c)$ is true (and $\mathsf{female}(c)$ is false) which contradicts the requirement of the second disjunct that $\mathsf{male}(c)$ is false. The inner disjunction $\mathsf{female}(c) \vee \neg\mathsf{female}(c)$ is obviously deterministic as well. Note that both disjunctions are also smooth. For the outer disjunction, this particularly holds because we appended the tautology $\mathsf{female}(c) \vee \neg\mathsf{female}(c)$. Thus, ϕ' is in

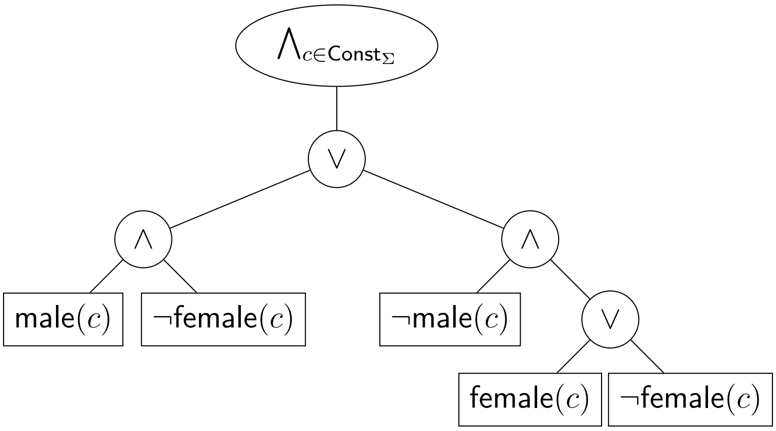

Figure 6.1: sd-DNNF-Circuit of the sentence ϕ' from Example 6.1.1.

sd-DNNF-normal form and, in order to indicate this, we could write

$$\phi' = \bigwedge\nolimits^{d}_{c\in\mathsf{Const}_\Sigma} \big((\mathsf{male}(c) \wedge^{d} \neg\mathsf{female}(c)) \\ \vee^{sd} (\neg\mathsf{male}(c) \wedge^{d} (\mathsf{female}(c) \vee^{sd} \neg\mathsf{female}(c)))\big).$$

Sentences which are in sd-DNNF-normal form can be depicted as so-called sd-DNNF-*circuits*. sd-DNNF-circuits are rooted directed acyclic graphs the inner nodes of which are either conjunctions or disjunctions, possibly with domain constraints, and the leaf nodes represent ground literals. We omit a formal definition of sd-DNNF-circuits but give an example which illustrates the very straightforward idea of sd-DNNF-circuits well.

Example 6.1.4
The sd-DNNF-circuit of the sentence ϕ' from Example 6.1.1 is shown in Figure 6.1.

Once a sentence $\phi \in \mathcal{RL}^S(\Sigma)$ is compiled into sd-DNNF-normal form, the counting of the models of ϕ can simply be done by translating the sentence into an arithmetic expression and evaluating this expression afterwards (cf. [Van den Broeck, 2013]). For this, every (decomposable) conjunction is replaced by a multiplication, every (deterministic and smooth) disjunction by an addition, and every ground literal by the constant 1. The resulting arithmetic expression equals the model count of ϕ. The process is formally captured by the so-called *count function* [Van den Broeck, 2013] which evaluates sentences in sd-DNNF-normal form according to the described arithmetic translation.

Definition 6.1.5: Count Function *[Van den Broeck, 2013]*

Let Σ be a finite signature. The *count function* $\mathsf{cnt}\colon \mathsf{sd\text{-}DNNF}(\Sigma) \to \mathbb{N}_0$ maps every sentence from $\mathsf{sd\text{-}DNNF}(\Sigma)$ to a natural number and is recursively defined as follows. The base cases of the count function are $\mathsf{cnt}(\top) = 1$, $\mathsf{cnt}(\bot) = 0$, and $\mathsf{cnt}(l) = 1$ for ground literals $l \in \mathsf{grLit}(\Sigma)$. Further, for every decomposable conjunction $\phi_1 \wedge^d \ldots \wedge^d \phi_m \in \mathsf{sd\text{-}DNNF}(\Sigma)$ and every smooth deterministic disjunction $\phi_1 \vee^{sd} \ldots \vee^{sd} \phi_m \in \mathsf{sd\text{-}DNNF}(\Sigma)$, $m \in \mathbb{N}$, one has

$$\mathsf{cnt}(\phi_1 \wedge^d \ldots \wedge^d \phi_m) = \prod_{i=1}^{m} \mathsf{cnt}(\phi_i),$$
$$\mathsf{cnt}(\phi_1 \vee^{sd} \ldots \vee^{sd} \phi_m) = \sum_{i=1}^{m} \mathsf{cnt}(\phi_i).$$

Note that in algebraic model counting the translation of sentences in sd-DNNF-normal form to algebraic statements works similar to the arithmetic translation in Definition 6.1.5. The difference in algebraic model counting is that literals are mapped to their algebraic weight instead of 1 and conjunctions and disjunctions are interpreted by the algebraic additive and multiplicative operators $\oplus$ and $\otimes$ (cf. Kimmig et al. [2017]).

For conjunctions and disjunctions with domain constraints, Definition 6.1.5 results in

$$\mathsf{cnt}\Big(\bigwedge\nolimits^{d}_{c \in \mathsf{Sol}(\mathsf{CS})} \phi\langle X/c\rangle\Big) = \prod_{c \in \mathsf{Sol}(\mathsf{CS})} \mathsf{cnt}(\phi\langle X/c\rangle),$$
$$\mathsf{cnt}\Big(\bigvee\nolimits^{d}_{c \in \mathsf{Sol}(\mathsf{CS})} \phi\langle X/c\rangle\Big) = \sum_{c \in \mathsf{Sol}(\mathsf{CS})} \mathsf{cnt}(\phi\langle X/c\rangle).$$

If for all $c_1, c_2 \in \mathsf{Sol}(\mathsf{CS})$ it holds that $\mathsf{cnt}(\phi\langle X/c_1\rangle) = \mathsf{cnt}(\phi\langle X/c_2\rangle)$, in addition, then the count of the conjunction respectively the disjunction simplifies to

$$\mathsf{cnt}\Big(\bigwedge\nolimits^{d}_{c \in \mathsf{Sol}(\mathsf{CS})} \phi\langle X/c\rangle\Big) = \mathsf{cnt}(\phi\langle X/c_1\rangle)^{|\mathsf{Sol}(\mathsf{CS})|},$$
$$\mathsf{cnt}\Big(\bigvee\nolimits^{d}_{c \in \mathsf{Sol}(\mathsf{CS})} \phi\langle X/c\rangle\Big) = |\mathsf{Sol}(\mathsf{CS})| \cdot \mathsf{cnt}(\phi\langle X/c_1\rangle).$$

The condition $\mathsf{cnt}(\phi\langle X/c_1\rangle) = \mathsf{cnt}(\phi\langle X/c_2\rangle)$ holds if $\phi\langle X/c_1\rangle$ and $\phi\langle X/c_2\rangle$ are *isomorphic*, i.e., if $\phi\langle X/c_1\rangle$ and $\phi\langle X/c_2\rangle$ are identical up to a renaming of constants. In order to indicate that the conjuncts of a conjunction respectively the disjuncts of a disjunction are isomorphic, we write $\wedge^i$ instead of $\wedge$ and $\vee^i$ instead of $\vee$.

Example 6.1.6

We obtained isomorphic conjuncts in the sentence ϕ' from Example 6.1.1 already. Actually, the subsentences ϕ_c, $c \in \mathsf{Const}_\Sigma$, in $\phi' = \bigwedge_{c \in \mathsf{Const}_\Sigma} \phi_c$ (cf. Example 6.1.3) are the same up to a renaming of the constant c: We can find a mapping $\rho\colon \mathsf{Const}_\Sigma \to \mathsf{Const}_\Sigma$ such that for every two constants $c_1, c_2 \in \mathsf{Const}_\Sigma$, $\rho(c_1) = c_2$ yields

$$\phi_{\rho(c_1)} = (\mathsf{male}(\rho(c_1)) \Rightarrow \neg\mathsf{female}(\rho(c_1))) = (\mathsf{male}(c_2) \Rightarrow \neg\mathsf{female}(c_2)) = \phi_{c_2}.$$

Hence, the outer conjunction in $\phi' = \bigwedge_{c \in \mathsf{Const}_\Sigma} \phi_c$ is isomorphic and we may write $\phi' = \bigwedge^{i}_{c \in \mathsf{Const}_\Sigma} \phi_c$.

The arithmetic translation of a sentence $\phi \in \mathsf{sd\text{-}DNNF}(\Sigma)$ (cf. Definition 6.1.5) can be depicted as a rooted directed acyclic graph, analogously to its sd-DNNF-circuit, which is called *counting graph* [Darwiche, 2001b]. In order to obtain the counting graph of a sentence in sd-DNNF one takes its sd-DNNF-circuit and replaces every leaf node by 1, every conjunction node by $\cdot$, and every disjunction node by $+$. If there is a node which represents a conjunction with domain constraint CS over isomorphic conjuncts, we write $(\ldots)^{|\mathsf{Sol}(\mathsf{CS})|}$ into the respective node in order to indicate that we may take the expression provided by the child nodes to the power of $|\mathsf{Sol}(\mathsf{CS})|$. Analogously, we write $|\mathsf{Sol}(\mathsf{CS})| \cdot (\ldots)$ when the node originates from a disjunction with domain constraint CS over isomorphic disjuncts.

Sometimes we also use a combination of sd-DNNF-circuits and counting graphs. Then, we write the arithmetic expressions as labels next to the corresponding nodes into the sd-DNNF-circuit of a sentence $\phi \in \mathsf{sd\text{-}DNNF}(\Sigma)$ in order to visualize both the sentence ϕ and the count function applied to ϕ in the same graph. In this representation form, we usually write the accumulated arithmetic expression next to a node, that is, we multiply or sum up all the arithmetic expressions of the child nodes as the next example illustrates.

Example 6.1.7

Figure 6.2 shows both the counting graph and the combined sd-DNNF-circuit with corresponding counts as labels for the sentence ϕ' from Example 6.1.1.

In [Van den Broeck, 2013] it is shown that count functions indeed count the models of sentences in sd-DNNF. We formalize this in the next proposition.

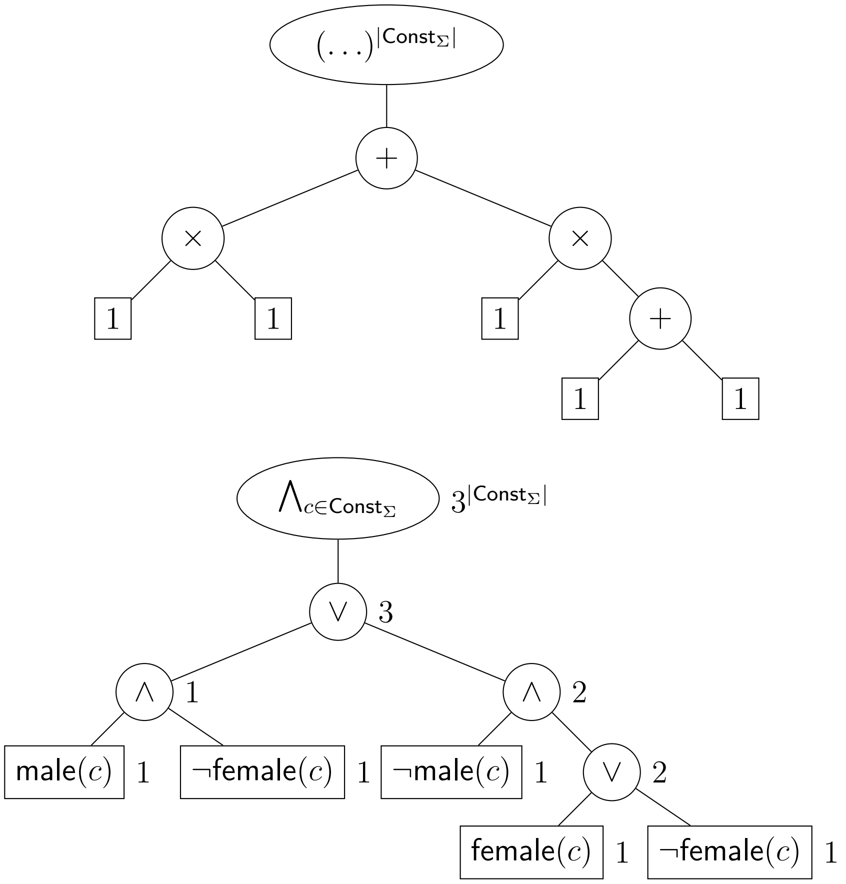

Figure 6.2: Counting graph (top) and sd-DNNF-circuit with the consolidated counts as labels (bottom) of the sentence ϕ' from Example 6.1.1.

Proposition 6.1.8: Correctness of Count Function
(cf. Proposition 3.14 in [Van den Broeck, 2013])

Let Σ be a finite signature, and let $\phi \in \mathsf{sd\text{-}DNNF}(\Sigma)$ be a sentence in sd-DNNF-normal form. Then,

$$\mathsf{cnt}(\phi) = \mathsf{FOMC}_\Sigma(\phi). \tag{6.8}$$

Proof. A proof of this proposition can be found in [Van den Broeck, 2013]. □

Note that the count function cnt does not need to mention the signature Σ as a subscript (cf. $\mathsf{cnt}(\phi)$ in (6.8)) because the count function is applied to sentences in sd-DNNF only and sentences ϕ in sd-DNNF yield $\mathsf{grAtom}(\Sigma) = \mathsf{grAtom}(\phi)$. Hence, the signature Σ is already clearly induced by the sentence, here by ϕ.

Sentences from $\mathcal{RL}^S(\Sigma)$ do not have a unique representation in sd-DNNF-normal form. For example, the sentence

$$\phi_c = (\mathsf{male}(c) \wedge^d \neg\mathsf{female}(c)) \vee^{sd} (\neg\mathsf{male}(c) \wedge^d (\mathsf{female}(c) \vee^{sd} \neg\mathsf{female}(c)))$$

from Example 6.1.3 is equivalent to

$$(\mathsf{male}(c) \wedge^d \neg\mathsf{female}(c)) \vee^{sd} (\neg\mathsf{male}(c) \wedge^d \mathsf{female}(c)) \vee^{sd} (\neg\mathsf{male}(c) \wedge^d \neg\mathsf{female}(c))$$

which is also in sd-DNNF. Basically, one can say that for efficient model counting of a sentence $\phi \in \mathcal{RL}^S(\Sigma)$, it is favorable to have an sd-DNNF-normal form representation of ϕ which mentions as few disjunctions as possible because disjunctions are translated to summations and executing sums is more expensive than executing multiplications when counting models. This is particularly the case because decomposable conjunctions ($\widehat{=}$ multiplications) reduce the model counting task to subproblems wrt. smaller subsignatures: Let $\phi = \phi_1 \wedge^d \ldots \wedge^d \phi_m$ be a sentence in sd-DNNF, and, for $i = 1, \ldots, m$, let Σ_i be the signature of ϕ_i. Then,

$$\mathsf{FOMC}_\Sigma(\phi) = \prod_{i=1}^{m} \mathsf{FOMC}_{\Sigma_i}(\phi_i).$$

The model counting tasks on the right-hand side of this equation refer to strict subsignatures of Σ (in case of $m > 1$) which reduces the computational complexity in a natural way.

▶ Compiling Sentences Into sd-DNNF-Normal Form

Now, we discuss some basic techniques on compiling any given sentence in $\mathcal{RL}^S(\Sigma)$ into sd-DNNF-normal form. More sophisticated techniques will be discussed in the context of *typed model counting* in Section 6.4. Developing powerful compiling strategies is of high importance in view of domain-lifted model counting. Here, we lay the focus on proving that any sentence from $\mathcal{RL}^S(\Sigma)$ can be compiled into sd-DNNF-normal form and do not claim that the presented techniques are the best to choose in the respective situations. Nevertheless, they demonstrate well the basic ideas behind the sd-DNNF-normal form.

For the following compilation of sentences $\phi \in \mathcal{RL}^S(\Sigma)$ into sd-DNNF-normal form, we assume that such a sentence ϕ is already made quantifier-free by replacing every quantification in ϕ by an equivalent conjunction or disjunction with domain constraints. After that, the first step to do is to bring ϕ into negation normal form. This can be achieved by recursively applying *De Morgan's laws*, for instance, i.e., by exploiting the logical equivalencies $\neg(\phi \vee \psi) \equiv \neg\phi \wedge \neg\psi$ and $\neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi$. We give an example for these precompilation steps.

Example 6.1.9
Let A and B be unary predicates. We bring the sentence

$$\phi = \neg\forall_{\mathsf{CS}} X.\ (A(X) \vee \overline{B(X)})$$

into negation normal form by first translating the universal quantification to a conjunction and then applying De Morgan's law:

$$\begin{aligned}\neg\forall_{\mathsf{CS}} X.\ (A(X) \vee \overline{B(X)}) &\equiv \neg \bigwedge_{c \in \mathsf{Sol}(\mathsf{CS})} (A(c) \vee \overline{B(c)}) \\ &\equiv \bigvee_{c \in \mathsf{Sol}(\mathsf{CS})} \neg(A(c) \vee \overline{B(c)}) \\ &\equiv \bigvee_{c \in \mathsf{Sol}(\mathsf{CS})} (\overline{A(c)} \wedge B(c)).\end{aligned}$$

The last expression is in negation normal form.

Now, we demonstrate how to make disjunctions deterministic by considering a disjunction of two disjuncts first. Let $\phi_1 \vee \phi_2 \in \mathcal{RL}^S(\Sigma)$. Then, there might be an interpretation which models both ϕ_1 and ϕ_2. In this case, $\phi_1 \vee \phi_2$ is not deterministic as per definition. In order to make the disjunction deterministic, we split the sentence $\phi_1 \vee \phi_2$ into the cases in which ϕ_1 holds but ϕ_2 does not hold, in which ϕ_2 holds but ϕ_1 does not hold, and in which both ϕ_1 and ϕ_2 hold. These three cases are mutually exclusive by construction, and

$$\phi_1 \vee \phi_2 \equiv (\phi_1 \wedge \overline{\phi_2}) \vee^d (\phi_1 \wedge \phi_2) \vee^d (\overline{\phi_1} \wedge \phi_2) \tag{6.9}$$

holds. Hence, when replacing $\phi_1 \vee \phi_2$ by the right-hand side of (6.9), all outer disjunctions are deterministic. For a disjunction with more than two disjuncts, any combination of possibly negated disjuncts can be built:

$$\phi_1 \vee \ldots \vee \phi_m \equiv \bigvee\nolimits^d_{\emptyset \subset S \subseteq \{1,\ldots,m\}} (\bigwedge_{i \in S} \phi_i \wedge \bigwedge_{i \in \{1,\ldots,m\} \setminus S} \overline{\phi_i}). \tag{6.10}$$

Note that the right-hand sides of (6.9) and (6.10) are not necessarily in negation normal form, even if the $\phi_i's$ are, because of the newly introduced negated subsentences $\overline{\phi_i}$. In order to re-establish the negation normal form, De Morgan's laws have to be applied again to these negated subsentences $\overline{\phi_i}$.

In order to make nested disjunctions in a sentence ϕ deterministic, one begins with applying the above-mentioned rewriting rules to the outermost disjunction $\phi_1 \vee \ldots \vee \phi_m$ and recursively proceeds with the disjunctions in the subsentences ϕ_i for $i \in \{1, \ldots, m\}$ and so on.

Example 6.1.10

Let Σ be a finite signature with $\mathsf{Const}_\Sigma = \{a, b, c\}$ and $\mathsf{Pred}_\Sigma = \{A/1, B/1\}$. We consider the sentence

$$\phi = \bigvee_{c \in \mathsf{Const}_\Sigma} (A(c) \vee B(c))$$

which is in negation normal form because there are no negations at all. However, the disjunctions in ϕ are not deterministic. Following the above-mentioned strategy which results in (6.10), we can make the outer disjunction deterministic by

$$\begin{aligned}
\phi = & \bigvee_{c \in \mathsf{Const}_\Sigma} (A(c) \vee B(c)) \\
\equiv & \quad (A(a) \vee B(a)) \wedge (A(b) \vee B(b)) \wedge (A(c) \vee B(c)) \\
& \vee^d \overline{(A(a) \vee B(a))} \wedge (A(b) \vee B(b)) \wedge (A(c) \vee B(c)) \\
& \vee^d (A(a) \vee B(a)) \wedge \overline{(A(b) \vee B(b))} \wedge (A(c) \vee B(c)) \\
& \vee^d (A(a) \vee B(a)) \wedge (A(b) \vee B(b)) \wedge \overline{(A(c) \vee B(c))} \\
& \vee^d \overline{(A(a) \vee B(a))} \wedge \overline{(A(b) \vee B(b))} \wedge (A(c) \vee B(c)) \\
& \vee^d \overline{(A(a) \vee B(a))} \wedge (A(b) \vee B(b)) \wedge \overline{(A(c) \vee B(c))} \\
& \vee^d (A(a) \vee B(a)) \wedge \overline{(A(b) \vee B(b))} \wedge \overline{(A(c) \vee B(c))}.
\end{aligned}$$

The next step is to bring the whole sentence in negation normal form again. This means, for example, replacing $\overline{(A(a) \vee B(a))}$ by $\overline{A(a)} \wedge \overline{B(a)}$ according to De Morgan's laws. Afterwards, the inner disjunctions have to be made deterministic and so on.

Note that it would be much easier here to count the models of $\neg\phi$ and to derive the model count of ϕ via $\mathsf{FOMC}_\Sigma(\phi) = |\mathsf{Int}_\Sigma| - \mathsf{FOMC}_\Sigma(\neg\phi)$. With this strategy, one obtains

$$\begin{aligned}
\neg\phi = \neg \bigvee_{c \in \mathsf{Const}_\Sigma} (A(c) \vee B(c)) \equiv \bigwedge_{c \in \mathsf{Const}_\Sigma} \neg(A(c) \vee B(c)) \equiv \bigwedge_{c \in \mathsf{Const}_\Sigma} (\overline{A(c)} \wedge \overline{B(c)}) \\
\equiv (\overline{A(a)} \wedge^d \overline{B(a)}) \wedge^d (\overline{A(b)} \wedge^d \overline{B(b)}) \wedge^d (\overline{A(c)} \wedge^d \overline{B(c)}),
\end{aligned}$$

and, hence, $\mathsf{FOMC}_\Sigma(\phi) = 2^6 - 1 = 31$. This example clearly shows that model counting techniques which are tailor-made for the specific sentence resp. situation can simplify the counting task drastically. On the other hand, there is not *the best* strategy which fits to all problems well.

In the next step, we focus on the conjunctions in sentences $\phi \in \mathcal{RL}^S(\Sigma)$. If a conjunction $\phi_1 \wedge \phi_2$ is mentioned in ϕ which is not decomposable, then ϕ_1 and ϕ_2 share at least one ground atom, i.e., there is $a \in \mathsf{grAtom}(\phi_1) \cap \mathsf{grAtom}(\phi_2)$. For the moment, we assume that a is the only element in the intersection of $\mathsf{grAtom}(\phi_1)$ and $\mathsf{grAtom}(\phi_2)$. In order to make $\mathsf{grAtom}(\phi_1)$ and $\mathsf{grAtom}(\phi_2)$ disjoint, one can apply the concept of *literal conditioning* [Darwiche, 2001b].

The basic idea behind *literal conditioning* is to condition a sentence on one of the two truth values "true" and "false" of a ground atom. Technically, one removes a ground atom from a sentence by replacing every occurrence of it by either $\top$ or $\bot$, depending on whether the ground atom shall be assumed to be true or false. By applying both replacements on the common ground atom a that is shared by ϕ_1 and ϕ_2 and by combining the two cases by a disjunction afterwards we observe an equivalent rewriting of $\phi_1 \wedge \phi_2$ the conjunctions of which are decomposable and which is known as the *Shannon decomposition* [Shannon, 1949]. More precisely, we proceed as follows. We write $\phi_i\langle a/\top\rangle$ for the replacement of every occurrence of a in ϕ_i by $\top$ and $\phi_i\langle a/\bot\rangle$ for the respective replacement by $\bot$. Then,

$$\phi_i \equiv a \wedge \phi_i\langle a/\top\rangle \vee \bar{a} \wedge \phi_i\langle a/\bot\rangle \tag{6.11}$$

is the Shannon decomposition of ϕ_i with respect to its ground atom $a \in \mathsf{grAtom}(\phi_i)$. The result may no longer be in negation normal form which is why we have to restore this property afterwards. In particular, we have to simplify $\neg\top \equiv \bot$ and $\neg\bot \equiv \top$. Alternatively, one can replace the literal $\neg a$ by $\top$ (or $\bot$) directly instead of replacing a by $\bot$ (or $\top$) first and fixing the expressions $\neg\bot$ (or $\neg\top$) afterwards whenever a negation stands in front of a. This justifies the name of literal conditioning. Because $a \vee \bar{a} \equiv \top$, $a \wedge \phi_i \equiv a \wedge \phi_i\langle a/\top\rangle$, and $\bar{a} \wedge \phi_i \equiv \bar{a} \wedge \phi_i\langle a/\bot\rangle$, one has

$$\begin{aligned}
\phi_1 \wedge \phi_2 &\equiv (a \vee^{sd} \bar{a}) \wedge \phi_1 \wedge \phi_2 \\
&\equiv (a \wedge \phi_1 \wedge \phi_2) \vee^{d} (\bar{a} \wedge \phi_1 \wedge \phi_2) \\
&\equiv (a \wedge (\phi_1 \wedge \phi_2)\langle a/\top\rangle) \vee^{d} (\bar{a} \wedge (\phi_1 \wedge \phi_2)\langle a/\bot\rangle) \\
&\equiv (a \wedge^{d} \phi_1\langle a/\top\rangle \wedge^{d} \phi_2\langle a/\top\rangle) \vee^{d} (\bar{a} \wedge^{d} \phi_1\langle a/\bot\rangle \wedge^{d} \phi_2\langle a/\bot\rangle).
\end{aligned}$$

Replacing $\phi_1 \wedge \phi_2$ by the expression on the right-hand side particularly ensures that the outer conjunctions between the replacements of ϕ_1 and ϕ_2 are decomposable when a was the only element in $\mathsf{grAtom}(\phi_1) \cap \mathsf{grAtom}(\phi_2)$. Also the newly introduced negation occurs directly in front of the ground atom a such that the resulting sentence is still in negation normal form. The newly introduced disjunction is obviously deterministic, as well.

If ϕ_1 and ϕ_2 share more than one ground atom, literal conditioning has to be applied with respect to every ground atom in $\mathsf{grAtom}(\phi_1) \cap \mathsf{grAtom}(\phi_2)$ to make $\phi_1 \wedge \phi_2$

decomposable. This can be done in sequence.

If the conjunction mentions more than two conjuncts, e.g., if it is a conjunction with domain constraints, literal conditioning can be applied to all conjuncts simultaneously:

$$\phi_1 \wedge \ldots \wedge \phi_m \equiv (a \wedge^d \bigwedge\nolimits^d_{i=1,\ldots,m} \phi_i\langle a/\top\rangle) \vee^d (\bar{a} \wedge^d \bigwedge\nolimits^d_{i=1,\ldots,m} \phi_i\langle a/\bot\rangle).$$

Eventually, by an iterative application of literal conditioning to all conjunctions in ϕ, beginning with the outermost conjunction $\phi_1 \wedge \ldots \wedge \phi_m$ in ϕ and then proceeding with the conjunctions within the single conjuncts ϕ_i, all conjunctions in ϕ can be successively made decomposable.

Example 6.1.11

Let Σ be a finite signature with $\mathsf{Pred}_\Sigma = \{A/0, B/1\}$. We consider the sentence

$$\phi = \bigwedge_{c\in\mathsf{Const}_\Sigma} (B(c) \Rightarrow A).$$

Applying literal conditioning on A leads to

$$\begin{aligned}
\phi &= \bigwedge_{c\in\mathsf{Const}_\Sigma} (B(c) \Rightarrow A) \\
&\equiv A \wedge \left(\bigwedge_{c\in\mathsf{Const}_\Sigma} (B(c) \Rightarrow A)\right)\langle A/\top\rangle \vee \overline{A} \wedge \left(\bigwedge_{c\in\mathsf{Const}_\Sigma} (B(c) \Rightarrow A)\right)\langle A/\bot\rangle \\
&\equiv A \wedge \bigwedge_{c\in\mathsf{Const}_\Sigma} (B(c) \Rightarrow \top) \vee \overline{A} \wedge \bigwedge_{c\in\mathsf{Const}_\Sigma} (B(c) \Rightarrow \bot) \\
&\equiv A \vee^d (\overline{A} \wedge^d \bigwedge\nolimits^d_{c\in\mathsf{Const}_\Sigma} \overline{B(c)}).
\end{aligned}$$

Finally, smoothness can be established as follows. If a disjunction $\phi_1 \vee \phi_2$ in a sentence ϕ is not smooth, then there is a ground atom $a \in \mathsf{grAtom}(\Sigma)$ such that either $a \in \mathsf{grAtom}(\phi_1) \setminus \mathsf{grAtom}(\phi_2)$ or $a \in \mathsf{grAtom}(\phi_2) \setminus \mathsf{grAtom}(\phi_1)$. Without loss of generality, we assume that $a \in \mathsf{grAtom}(\phi_1) \setminus \mathsf{grAtom}(\phi_2)$ holds and replace ϕ_2 by $\phi_2^{\mathsf{new}} = \phi_2 \wedge^d (a \vee^{sd} \bar{a})$. Because $(a \vee^{sd} \bar{a})$ is a tautology, ϕ_2 and ϕ_2^{new} are logically equivalent. This procedure can be repeated until $\mathsf{grAtom}(\phi_1^{\mathsf{new}}) = \mathsf{grAtom}(\phi_2^{\mathsf{new}})$, where ϕ_1^{new} stands for ϕ_1 after applying the replacement step with respect to all ground atoms in $\mathsf{grAtom}(\phi_2) \setminus \mathsf{grAtom}(\phi_1)$ (ϕ_2^{new} analogously). $\phi_1^{\mathsf{new}} \vee^s \phi_2^{\mathsf{new}}$ is smooth, in negation normal form if ϕ_1^{new} and ϕ_2^{new} are in negation normal form, and the newly introduced disjunctions are deterministic and the conjunctions decomposable. The handling is the same for disjunctions with more than two disjuncts as the disjuncts can be smoothed pairwise. Eventually, the whole procedure can be applied to all conjunctions in the sentence one after the other such that all conjunctions in ϕ are made smooth. In the end, one has to take care of the requirement $\mathsf{grAtom}(\phi) = \mathsf{grAtom}(\Sigma)$. If this condition is not fulfilled, then ϕ is replaced by

$$\phi \wedge^d \bigwedge\nolimits^d_{a \in \mathsf{grAtom}(\Sigma) \setminus \mathsf{grAtom}(\phi)} (a \vee^{sd} \bar{a})$$

in order to add the ground atoms "missing" in ϕ.

Example 6.1.12
We recall the sentence ϕ from Example 6.1.11 which was compiled in Example 6.1.11 into

$$\phi' = A \vee^d (\overline{A} \wedge^d \bigwedge\nolimits^d_{c \in \mathsf{Const}_\Sigma} \overline{B(c)}).$$

The sentence ϕ' is not in sd-DNNF because the disjunction in ϕ' is not smooth. We can establish the smoothness of ϕ' by adding the tautologies $(B(c) \vee \overline{B(c)})$ for all $c \in \mathsf{Const}_\Sigma$ to the first disjunct in ϕ' such that ϕ' becomes

$$\phi' \equiv (A \wedge^d \bigwedge\nolimits^d_{c \in \mathsf{Const}_\Sigma} (B(c) \vee^{sd} \overline{B(c)})) \vee^{sd} (\overline{A} \wedge^d \bigwedge\nolimits^d_{c \in \mathsf{Const}_\Sigma} \overline{B(c)})$$

which is now in sd-DNNF. The model count of ϕ' is

$$\mathsf{FOMC}_\Sigma(\phi') = (1 \cdot (1+1)^{|\mathsf{Const}_\Sigma|}) + (1 \cdot (1)^{|\mathsf{Const}_\Sigma|}) = 2^{|\mathsf{Const}_\Sigma|} + 1.$$

In the next section, we extend first-order model counting to *typed model counting* and the sd-DNNF-normal form to so-called *structured sentences*.

6.2 Typed Model Counting

In this section, we develop with *typed model counting* [Wilhelm and Kern-Isberner, 2017; Wilhelm et al., 2017a, 2019b] a variant of first-order model counting with the aim not only to count the models of a first-order sentence but also to distinguish its models with respect to their *types*. The basic idea behind these types is to reflect which parts of the sentence are satisfied by the model. Therewith, one can read from the model type in some sense *why* the sentence is satisfied by the model.

The types are constituted by so-called *structure elements* which are directly written into the sentences. As a consequence, typed model counting requires that the sentences are enriched by these structure elements and, therefore, typed model counting is not defined for sentences in $\mathcal{RL}^S(\Sigma)$ directly but for so-called *structured sentences* from a "structured extension" of $\mathcal{RL}^S(\Sigma)$. However, note that counting the typed models of a structured sentence can be compiled into an algebraic model counting task with respect to a classical sentence in $\mathcal{RL}^S(\Sigma)$ (cf. Section 6.3).

ω	$[\phi]_\omega$	type of ϕ_S	ω	$[\phi]_\omega$	type of ϕ_S
CMP	1	$s_1 s_2$	$CM\overline{P}$	1	s_1
$\overline{C}MP$	1	s_2	$\overline{C}M\overline{P}$	0	0
$C\overline{M}P$	1	s_1	$C\overline{M}\overline{P}$	1	s_1
$\overline{C}\overline{M}P$	0	0	$\overline{C}\overline{M}\overline{P}$	0	0

Table 6.1: Evaluation of ϕ and ϕ_S from Example 6.2.1.

▶ Motivation of Typed Model Counting

Our motivation of developing typed model counting is to provide a formal framework for a systematic computation of conditional structures of knowledge bases. The basic idea is to compile conditionals into structured sentences such that their typed models correspond to the conditional structures of possible worlds with respect to the initial conditionals. The benefit of this approach is that we have recourse to efficient first-order model counting techniques when computing conditional structures.

The power of typed model counting lies in the fact that structure elements can be combined to complex types in the same ways as the verifications and falsifications of the single instances of open conditionals aggregate to the conditional structure of the whole conditional. To illustrate how typed model counting works in general, we give a simple example detached from the application to conditional reasoning.

Example 6.2.1

We consider the signature $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ with $\mathsf{Const}_\Sigma = \emptyset$ and $\mathsf{Pred}_\Sigma = (\{C/0, M/0, P/0\}$, where the predicates C, M, and P stand for "computer scientist," "mathematician," and "passion for computer science," respectively. The expression

$$\phi_S = (s_1 \circ C) \vee (s_2 \circ (M \wedge P))$$

> "Someone is a computer scientist or a mathematician with a passion for computer science."

can be understood as a structured sentence with the structure elements s_1 and s_2. If we ignore the structure elements for a while, we obtain the classical sentence

$$\phi = C \vee (M \wedge P).$$

The evaluation of ϕ is shown in Table 6.1. We can see that the sentence is true in those possible worlds in which, informally spoken, the imagined person is a computer scientist, or a mathematician with a passion for computer science. What we do not gain from the interpretations is how the satisfaction of the sentence depends on the satisfaction of its subsentences, that is, we are not told whether the person is a computer scientist, a mathematician with a passion for computer science, or maybe both. The idea behind typed model counting is to distinguish among these cases, as discussed next.

Whenever C holds, then the structured sentence ϕ_S is not just true, but the structure element s_1, which is annotated to C, is multiplied to the truth value 1. If in addition M and P hold, that is, if the possible world is $\omega = CMP$, then s_2 is incorporated into the interpretation of ϕ_S as well, and we obtain that ω is a model of ϕ_S of type $s_1 s_2$. If M or P would not hold, then the model type would be s_1 and, hence, we can read from the evaluation of ϕ_S why ϕ_S is true (or not). The complete evaluation of ϕ_S is also shown in Table 6.1. We gain the information that there is one model of ϕ_S in which the person is a mathematician with a passion for computer science (s_2), there are three models of ϕ_S in which the person is a computer scientist (s_1), and there is one model in which the person is both ($s_1 s_2$). Of course, the placement of the structure elements matters. It determines regarding what the models can be differentiated.

As we have seen, the formal basis of typed model counting is a structured language which we define next.

▶ Structured Language $\mathcal{SL}(\Sigma)$

The *structured language* $\mathcal{SL}(\Sigma)$ constitutes the formal framework for typed model counting and combines the relational language $\mathcal{RL}(\Sigma)$ with a free Abelian semiring $\mathcal{S}$. For a definition of free Abelian semirings, please see Definition A.1.4 in the Appendix.

Definition 6.2.2: Structured Language *(cf. [Wilhelm et al., 2017a])*

Let $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ be a finite signature, let $\mathcal{RL}(\Sigma)$ be a relational language, and let $\mathcal{S} = (\mathcal{S}_\mathfrak{G}, \oplus, \otimes, 0, 1)$ be a free Abelian semiring with generating set $\mathcal{S}_\mathfrak{G}$. Then, the *structured language* $\mathcal{SL}(\Sigma)$ is the smallest set such that

$$\phi, \quad \psi_S \wedge \chi_S, \quad \psi_S \vee \chi_S, \quad \forall_{\mathsf{CS}} X.\psi_S, \quad \exists_{\mathsf{CS}} X.\psi_S, \quad s \circ \psi_S \tag{6.12}$$

are in $\mathcal{SL}(\Sigma)$ where $\phi \in \mathcal{RL}(\Sigma)$, $\psi_S, \chi_S \in \mathcal{SL}(\Sigma)$, $X \in \mathsf{FreeVar}(\psi)$, $s \in \mathcal{S}_\mathfrak{G}$, and $\circ\colon \mathcal{S} \times \mathcal{SL}(\Sigma) \to \mathcal{SL}(\Sigma)$ is an operation between $\mathcal{S}$ and $\mathcal{SL}(\Sigma)$. The operation $\circ$ shall bind stronger than all connectives in $\mathcal{RL}(\Sigma)$ so that we can omit brackets. The elements in $\mathcal{SL}(\Sigma)$ are called *structured formulas* and the elements in $\mathcal{S}_{\mathfrak{G}}$ are *structure elements*. Structured formulas without free variables are *structured sentences*. We denote the set of all structured sentences with $\mathcal{SL}^S(\Sigma)$.

In plain words, the structured language $\mathcal{SL}(\Sigma)$ consists of the relational formulas from $\mathcal{RL}(\Sigma)$ (cf. "ϕ" in (6.12)) to which structure elements can be concatenated from the left ("$s \circ \psi_S$" in (6.12)). Further, structured formulas can be compounded by the common connectives from $\mathcal{RL}(\Sigma)$ except for negation, i.e., by $\wedge$, $\vee$, $\forall$, and $\exists$. The operation $\circ$ is specified by the interpretations of structured sentences.

Note that negations do not occur in (6.12) unless they are part of ϕ, ψ_S, or χ_S. This ensures that structure elements are not in the scope of any negation. The reason why we do not allow negations of structure elements is the following: In contrast to classical logics in which we have the two truth values 0 and 1, we want to "split" the truth value 1 into several truth values (the *model types*) here, namely the elements from $\mathcal{S}_{\mathfrak{G}}$. While in classical logics it is clear that the negation of a sentence which does not hold is true (and vice versa), this bivalence principle does not hold in $\mathcal{SL}(\Sigma)$ as interpretations are many-valued here. Intuitively, the truth value of the negation of a false structured sentence with negation in front of structure elements could be any element from $\mathcal{S}_{\mathfrak{G}}$ and, hence, would not be well-defined.

We usually do not denote the semiring $\mathcal{S}$ as an argument to $\mathcal{SL}(\Sigma)$ and $\mathcal{SL}^S(\Sigma)$ because it will always be clear from the context which semiring is used.

Example 6.2.3

The structured language which is used in Example 6.2.1 is $\mathcal{SL}(\Sigma)$ with $\Sigma = (\emptyset, \{C/0, M/0, F/0\})$ and $\mathcal{S} = (\{s_1, s_2\}, +, \cdot, 0, 1)$. The expression ϕ_S from Example 6.2.1 is indeed a structured sentence according to Definition 6.2.2. In particular, there is no negation in ϕ_S in front of s_1 or s_2. In contrast to that the expression

$$\psi_S = \neg(s_1 \circ C \vee s_2 \circ M)$$

is not a structured sentence because both s_1 and s_2 are within the scope of the negation.

Formally, the semantics of structured sentences is given by *structured interpretations*.

Definition 6.2.4: Structured Interpretation *(cf. [Wilhelm et al., 2017a])*

Let $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ be a finite signature, and let $\mathcal{SL}(\Sigma)$ be a structured language with respect to a free Abelian semiring $\mathcal{S} = (\mathcal{S}_{\mathfrak{G}}, \oplus, \otimes, 0, 1)$. Then, a mapping $I_S \colon \mathcal{SL}^S(\Sigma) \to \mathcal{S}_{\mathfrak{G}} \cup \{0\}$ from the set of structured sentences to the structure elements in $\mathcal{S}_{\mathfrak{G}}$ conjoint with 0 is a *structured interpretation* iff it maps every ground atom from $\mathsf{grAtom}(\Sigma)$ to either 0 or 1 and is recursively defined for compounded structured sentences from $\mathcal{SL}^S(\Sigma)$ by

- $I_{\mathcal{S}}(\top) = 1$ and $I_{\mathcal{S}}(\bot) = 1$,
- $I_{\mathcal{S}}(\neg\phi) = \begin{cases} 1 & \text{if} \quad I_{\mathcal{S}}(\phi) = 0 \\ 0 & \text{otherwise} \end{cases}$,
- $I_{\mathcal{S}}(\psi_S \wedge \psi'_S) = I_{\mathcal{S}}(\psi_S) \otimes I_{\mathcal{S}}(\psi'_S)$,
- $I_{\mathcal{S}}(\psi_S \vee \psi'_S) = \begin{cases} I_{\mathcal{S}}(\psi_S) & \text{if} \quad I_{\mathcal{S}}(\psi'_S) = 0, \\ I_{\mathcal{S}}(\psi'_S) & \text{if} \quad I_{\mathcal{S}}(\psi_S) = 0, \\ I_{\mathcal{S}}(\psi_S) \otimes I_{\mathcal{S}}(\psi'_S) & \text{otherwise} \end{cases}$,
- $I_{\mathcal{S}}(s \circ \psi_S) = s \otimes I_{\mathcal{S}}(\psi_S)$,
- $I_{\mathcal{S}}(\exists_{\mathsf{CS}} X.\chi_S) = I_{\mathcal{S}}(\bigvee_{c \in \mathsf{Sol}_{\Sigma}(\mathsf{CS})} \chi_S\langle X/c\rangle)$,
- $I_{\mathcal{S}}(\forall_{\mathsf{CS}} X.\chi_S) = I_{\mathcal{S}}(\bigwedge_{c \in \mathsf{Sol}_{\Sigma}(\mathsf{CS})} \chi_S\langle X/c\rangle)$,

where $\phi \in \mathcal{RL}^S(\Sigma)$, $\psi_S, \psi'_S \in \mathcal{SL}^S(\Sigma)$, $\chi_S \in \mathcal{SL}(\Sigma)$ with $\mathsf{FreeVar}(\chi_S) = \{X\}$, and $s \in \mathcal{S}_{\mathfrak{G}}$.[a]

---

[a] Recall that $\chi_S\langle X/c\rangle$ means the syntactical substitution of the variable X by the constant c in χ_S.

Note that the structured interpretation of disjunctions $I_{\mathcal{S}}(\psi_S \vee \psi'_S)$ is well-defined although the cases in the definition are not exclusive (cf. the third item in the list in Definition 6.2.4). If both conditions $I_{\mathcal{S}}(\psi'_S) = 0$ and $I_{\mathcal{S}}(\psi_S) = 0$ are true, then the first two cases apply. However, this is not problematic because in both cases the disjunction is interpreted to 0.

Furthermore, the structured interpretation of disjunctions $I_{\mathcal{S}}(\psi_S \vee \psi'_S)$ is in line with the interpretation of disjunctions in relational languages. Although the interpretation of disjunctions in relational languages is often associated with sums, especially in the context of model counting—recall that $I(\phi \vee \psi) = I(\phi) + I(\psi)$ holds if ϕ and ψ are mutually exclusive sentences from $\mathcal{RL}^S(\Sigma)$ (cf. Proposition 2.2.8)—, the interpretation of disjunctions in $\mathcal{RL}(\Sigma)$ can also be defined in the same fashion as in Definition 6.2.4. We prefer the formulation that is used in Definition 6.2.4 here, because it intuitively shows that for every structured interpretation $I_{\mathcal{S}}$ the co-domain of $I_{\mathcal{S}}$ is a product of elements from $\mathcal{S}_{\mathfrak{G}}$ or 0. The summation in $\mathcal{S}$ is reserved for the task of counting the *typed models* which we will introduce in Definition 6.2.8.

With $\mathsf{Int}_{\mathcal{S}}(\Sigma)$ we denote the set of all structured interpretations over Σ. The set $\mathsf{Int}_{\mathcal{S}}(\Sigma)$ is in a one-to-one correspondence with $\mathsf{Int}(\Sigma)$ as both structured interpretations and classical interpretations depend in their evaluation of (structured) sentences on the assignment to the ground atoms in $\mathsf{grAtom}(\Sigma)$ only. Once this assignment is fixed, both the corresponding interpretation and the corresponding structured interpretation are fixed as well. In particular, structured interpretations interpret sentences without structure elements the same as their classical counterparts.

Because structured interpretations are in a one-to-one correspondence to interpretations and interpretations are in a one-to-one correspondence to possible worlds in this thesis, we can also identify the structured interpretations in $\mathsf{Int}_{\mathcal{S}}(\Sigma)$ with the possible worlds in $\Omega(\Sigma)$ when respecting the evaluation of the ground atoms in $\mathsf{grAtom}(\Sigma)$. We denote the structured interpretation which corresponds to $\omega \in \Omega(\Sigma)$ with $I^{\omega}_{\mathcal{S}}$.

Example 6.2.5
The formal evaluation of the sentence ϕ_S from Example 6.2.1 over the free Abelian semiring $\mathcal{S} = (\{s_1, s_2\}, +, \cdot, 0, 1)$ is as follows. Let $I_{\mathcal{S}}$ be a structured interpretation. Then,

$$\begin{aligned}
I_{\mathcal{S}}(\phi_S) &= I_{\mathcal{S}}((s_1 \circ C) \vee (s_2 \circ (M \wedge P))) \\
&= \begin{cases} I_{\mathcal{S}}(s_1 \circ C) & \text{if} \quad I_{\mathcal{S}}(s_2 \circ (M \wedge P)) = 0 \\ I_{\mathcal{S}}(s_2 \circ (M \wedge P)) & \text{if} \quad I_{\mathcal{S}}(s_1 \circ C) = 0 \\ I_{\mathcal{S}}(s_1 \circ C) \cdot I_{\mathcal{S}}(s_2 \circ (M \wedge P)) & \text{otherwise} \end{cases} \\
&= \begin{cases} s_1 \cdot I_{\mathcal{S}}(C) & \text{if} \quad s_2 \cdot I_{\mathcal{S}}(M \wedge P) = 0 \\ s_2 \cdot I_{\mathcal{S}}(M \wedge P) & \text{if} \quad s_1 \cdot I_{\mathcal{S}}(C) = 0 \\ s_1 \cdot I_{\mathcal{S}}(C) \cdot s_2 \cdot I_{\mathcal{S}}(M \wedge P) & \text{otherwise} \end{cases} \\
&= \begin{cases} s_1 \cdot I_{\mathcal{S}}(C) & \text{if} \quad I_{\mathcal{S}}(M) \cdot I_{\mathcal{S}}(P) = 0 \\ s_2 \cdot I_{\mathcal{S}}(M) \cdot I_{\mathcal{S}}(P) & \text{if} \quad I_{\mathcal{S}}(C) = 0 \\ s_1 \cdot I_{\mathcal{S}}(C) \cdot s_2 \cdot I_{\mathcal{S}}(M) \cdot I_{\mathcal{S}}(P) & \text{otherwise} \end{cases}
\end{aligned}$$

Given the possible world $\omega = \overline{C}MF$, i.e., $I^{\omega}_{\mathcal{S}}(C) = 0$ and $I^{\omega}_{\mathcal{S}}(M) = I\omega_{\mathcal{S}}(P) = 1$, for instance, we have $I^{\omega}_{\mathcal{S}}(\phi_S) = s_2$.

We have the following proposition which deepens the relationship between structured interpretations and classical interpretations.

ω	$I_{\mathcal{S}}^{\omega}(\phi_S)$	$I_{\mathcal{S}}^{\omega}(\phi)$	ω	$I_{\mathcal{S}}^{\omega}(\phi_S)$	$I_{\mathcal{S}}^{\omega}(\phi)$
ABC	s_1s_2	1	$\overline{A}BC$	0	0
$AB\overline{C}$	s_1s_2	1	$\overline{A}B\overline{C}$	0	0
$A\overline{B}C$	s_1	1	$\overline{A}\overline{B}C$	0	0
$A\overline{B}\overline{C}$	0	0	$\overline{A}\overline{B}\overline{C}$	0	0

Table 6.2: Structured interpretations of the sentences ϕ_S and ϕ from Example 6.2.7.

Proposition 6.2.6: Structured Versus Classical Interpretations

Let Σ be a finite signature, let $\phi_S \in \mathcal{SL}^S(\Sigma)$ be a structured sentence, and let $\phi \in \mathcal{RL}(\Sigma)$ be the sentence obtained from ϕ_S by canceling out all structure elements from ϕ_S, i.e., every subsentence $s \circ \psi$ in ϕ_S with $s \in \mathcal{S}_{\mathfrak{G}}$ and $\psi \in \mathcal{SL}^S(\Sigma)$ is replaced by ψ. Then, for every structured interpretation $I_S \in \mathsf{Int}_{\mathcal{S}}(\Sigma)$ it holds that $I_{\mathcal{S}}(\phi_S) = 0$ iff $I(\phi) = 0$ where I is the corresponding interpretation in $\mathsf{Int}(\Sigma)$.

Proof. This is a simple observation when comparing the definition of interpretations (Definition 2.2.7) and the definition of structured interpretations (Definition 6.2.4). The only case in which they differ and which introduces structure elements from $\mathcal{S}_{\mathfrak{G}}$ into structured interpretations is the case $s \circ \psi_S$ for which no pendant in the classical sense exists. However, this case does not apply when there is no structured element written into the formula as it is the case in ϕ. We just have to identify the identity elements 0 and 1 of the semiring $\mathcal{S}$ with the truth values 0 and 1 of classical interpretations as well as $\oplus$ and $\otimes$ with the classical addition and multiplication.

Also note that $s_1 \otimes s_2 \neq 0$ for all $s_1, s_2 \in \mathcal{S}_{\mathfrak{G}}$ holds. If $s_1 \otimes s_2 = 0$ would be true, then there would be $s_3 \in \mathcal{S}_{\mathfrak{G}}$ with $s_1 \otimes s_2 \otimes s_3 = 0 = s_1 \otimes s_2 \otimes s_3 \otimes s_3$ which contradicts the fact that every element in $\mathcal{S}$ can be generated by $\mathcal{S}_{\mathfrak{G}}$ in a unique way. □

Due to Proposition 6.2.6, $I_{\mathcal{S}}(\phi_S) = 0$ can only happen if the logical parts in ϕ_S are not satisfied by $I_{\mathcal{S}}$. The reason for this is that 0 is an element of the semiring $\mathcal{S}$ but not of $\mathcal{S}_{\mathfrak{G}}$. Thus, 0 does not occur in ϕ_S as a prefactor. Further, 0 cannot be generated by a multiplication of structure elements from $\mathcal{S}_{\mathfrak{G}}$ due to the definition of generators of free Abelian semirings. We give an example.

Example 6.2.7

We consider the signature $\Sigma = (\emptyset, \{A/0, B/0, C/0\})$ and the structured sentence

$$\phi_S = s_1 \circ A \wedge (s_2 \circ B \vee C)$$

with the structure elements s_1 and s_2. Then, its classical pendant is

$$\phi = A \wedge (B \vee C).$$

The evaluations of both ϕ_S and ϕ in the structured interpretations $I_{\mathcal{S}}^{\omega} \in \mathsf{Int}_{\mathcal{S}}(\Sigma)$ are shown in Table 6.2. The structured sentence ϕ_S is interpreted to 0 in the same structured interpretations as the sentence ϕ.

Proposition 6.2.6 shows that structured interpretations generalize classical interpretations. They indeed differ from classical interpretations only in that they collect all structure elements that are concatenated to the left of the satisfied parts of the structured sentence (see the definition of $I_{\mathcal{S}}(s \circ \psi_S)$ in Definition 6.2.4).

▶ Typed Models

We introduce the central notion of *typed models* which are the structured pendants to models of classical sentences.

Definition 6.2.8: Typed Model *(cf. [Wilhelm et al., 2017a])*

Let Σ be a finite signature, let $\phi_S \in \mathcal{SL}^S(\Sigma)$ be a structured sentence, and let $I_{\mathcal{S}} \in \mathsf{Int}_{\mathcal{S}}(\Sigma)$ be a structured interpretation. If $I_{\mathcal{S}}(\phi_S) \neq 0$, then $I_{\mathcal{S}}$ is a *typed model* of ϕ_S. More precisely, if $I_{\mathcal{S}}(\phi_S) = s$ with $s \neq 0$, then $I_{\mathcal{S}}$ is a *model of type* s of ϕ_S. Otherwise, if $I_{\mathcal{S}}(\phi_S) = 0$, then ϕ_S is not satisfied in $I_{\mathcal{S}}$.

If $I_{\mathcal{S}}$ is a typed model of ϕ_S, regardless of the specific type, i.e., if $I_{\mathcal{S}}(\phi_S) \neq 0$, then we write $I_{\mathcal{S}} \models_{\Sigma} \phi_S$. With

$$\mathsf{Mod}_{\Sigma}^{\mathcal{S}}(\phi_S) = \{I_{\mathcal{S}} \in \mathsf{Int}_{\mathcal{S}}(\Sigma) \mid I_{\mathcal{S}} \models_{\Sigma} \phi_S\}$$

we denote the set of all typed models of ϕ_S wrt. the signature Σ. However, usually we are interested not only in the information which structured interpretations model a structured sentence but also in the different types of the models. This is reflected in the following *typed model counting task.*

Definition 6.2.9: Typed Model Counting *(cf. [Wilhelm et al., 2017a])*

Let Σ be a finite signature, and let $\mathcal{SL}(\Sigma)$ be a structured language. Then, *typed model counting* is the task of counting all model types of a structured sentence $\phi_S \in \mathcal{SL}^S(\Sigma)$, i.e., the task of computing

$$\mathsf{TMC}_\Sigma(\phi_S) = \bigoplus_{I_\mathcal{S} \in \mathsf{Mod}^\mathcal{S}_\Sigma(\phi_S)} I_\mathcal{S}(\phi_S).$$

The result of typed model counting is an element in the underlying semiring $\mathcal{S}$. If $\mathcal{S}$ is zerosumfree (cf. (3.4)), then the number of summands in $\mathsf{TMC}_\Sigma(\phi_S)$ matches the number of typed models of ϕ_S which justifies the term *count*. We say that two structured sentences $\phi_S, \psi_S \in \mathcal{SL}^S(\Sigma)$ are $\mathcal{S}$-*equivalent* iff they agree on their model type in all structured interpretations.

Definition 6.2.10: $\mathcal{S}$-Equivalence *(cf. [Wilhelm et al., 2017a])*

Let Σ be a finite signature, and let $\phi_S, \psi_S \in \mathcal{SL}^S(\Sigma)$ be structured sentences. Then, ϕ_S and ψ_S are $\mathcal{S}$-*equivalent*, in symbols

$$\phi_S \equiv_\mathcal{S} \psi_S, \quad \text{iff} \quad I_S(\phi_S) = I_S(\psi_S)$$

for all $I_S \in \mathsf{Int}_\mathcal{S}(\Sigma)$.

The $\mathcal{S}$-equivalence of ϕ_S and ψ_S implies $\mathsf{TMC}_\Sigma(\phi_S) = \mathsf{TMC}_\Sigma(\psi_S)$. $\mathcal{S}$-equivalence is a stronger notion than logical equivalence because ϕ_S and ψ_S not only have to have the same typed models in order to be $\mathcal{S}$-equivalent but the counts of the models of every single type have to be the same, too.

Example 6.2.11
It is easy to see that the structured sentence

$$\psi_S = (s_1 \circ C \vee \overline{C}) \wedge (s_2 \circ (M \wedge P) \vee ((\overline{M} \wedge (P \vee \overline{P})) \vee (M \wedge \overline{P})))$$

is $\mathcal{S}$-equivalent to the sentence ϕ_S from Example 6.2.1.

Note that many calculation rules which preserve logical equivalence of sentences in $\mathcal{RL}^S(\Sigma)$ carry over to $\mathcal{S}$-equivalence of sentences in $\mathcal{SL}^S(\Sigma)$. We mention the next proposition to name some important calculation rules.

Proposition 6.2.12: Calculation Rules for Structured Sentences
Let Σ be a finite signature, and let $\phi_S, \psi_S, \chi_S \in \mathcal{SL}^S(\Sigma)$. Then,

- $\phi_S \wedge \psi_S \equiv_{\mathcal{S}} \psi_S \wedge \phi_S$ and $\phi_S \vee \psi_S \equiv_{\mathcal{S}} \psi_S \vee \phi_S$, (commutativity)
- $\phi_S \wedge (\psi_S \wedge \chi_S) \equiv_{\mathcal{S}} (\phi_S \wedge \psi_S) \wedge \chi_S$ and $\phi_S \vee (\psi_S \vee \chi_S) \equiv_{\mathcal{S}} (\phi_S \vee \psi_S) \vee \chi_S$, (associativity)
- $\phi_S \wedge \top \equiv_{\mathcal{S}} \phi_S$ and $\phi_S \vee \bot \equiv_{\mathcal{S}} \phi_S$. (neutrality)

Proof. Commutativity: The commutativity of $\equiv_{\mathcal{S}}$ follows from the commutativity of $\otimes$ in the semiring $\mathcal{S}$. For an arbitrary structured interpretation $I_{\mathcal{S}} \in \mathsf{Int}_{\mathcal{S}}(\Sigma)$, it holds that

$$I_{\mathcal{S}}(\phi_S \wedge \psi_S) = I_{\mathcal{S}}(\phi_S) \otimes I_{\mathcal{S}}(\psi_S) = I_{\mathcal{S}}(\psi_S) \otimes I_{\mathcal{S}}(\phi_S) = I_{\mathcal{S}}(\psi_S \wedge \phi_S).$$

The proof with respect to disjunction is analogous.

Associativity: The associativity of $\equiv_{\mathcal{S}}$ follows from the associativity of $\otimes$ in the semiring $\mathcal{S}$. Again, we give the proof for conjunctions, the proof for disjunctions is analogous. Let $I_{\mathcal{S}} \in \mathsf{Int}_{\mathcal{S}}(\Sigma)$. Then,

$$\begin{aligned} I_{\mathcal{S}}(\phi_S \wedge (\psi_S \wedge \chi_S)) &= I_{\mathcal{S}}(\phi_S) \otimes I_{\mathcal{S}}(\psi_S \wedge \chi_S) = I_{\mathcal{S}}(\phi_S) \otimes I_{\mathcal{S}}(\psi_S) \otimes I_{\mathcal{S}}(\chi_S) \\ &= I_{\mathcal{S}}(\phi_S \wedge \psi_S) \otimes I_{\mathcal{S}}(\chi_S) = I_{\mathcal{S}}((\phi_S \wedge \psi_S) \wedge \chi_S). \end{aligned}$$

Neutrality: Let $I_{\mathcal{S}} \in \mathsf{Int}_{\mathcal{S}}(\Sigma)$. Because $I_{\mathcal{S}}(\top) = 1$ and $I_{\mathcal{S}}(\bot) = 0$, we have

$$I_{\mathcal{S}}(\phi_S \wedge \top) = I_{\mathcal{S}}(\phi_S) \otimes I_{\mathcal{S}}(\top) = I_{\mathcal{S}}(\phi_S) \otimes 1 = I_{\mathcal{S}}(\phi_S).$$

Also,

$$\begin{aligned} I_{\mathcal{S}}(\phi_S \vee \bot) &= \begin{cases} I_{\mathcal{S}}(\phi_S) & \text{if} \quad I_{\mathcal{S}}(\bot) = 0 \\ I_{\mathcal{S}}(\bot) & \text{if} \quad I_{\mathcal{S}}(\phi_S) = 0 \\ I_{\mathcal{S}}(\phi_S) \otimes I_{\mathcal{S}}(\bot) & \text{otherwise} \end{cases} \\ &= I_{\mathcal{S}}(\phi_S) \end{aligned}$$

because the first case of the case differentiation applies (and perhaps the second case, namely if $I_{\mathcal{S}}(\phi_S) = I_{\mathcal{S}}(\bot) = 0$; then both cases yield the same result). □

Some other calculation rules do not hold in $\mathcal{SL}(\Sigma)$, like *distributivity* and *absorption.* The main reason for this is that *idempotence* is violated, i.e., in general

$$I_{\mathcal{S}}(\phi_S \wedge \phi_S) = I_{\mathcal{S}}(\phi_S) \otimes I_{\mathcal{S}}(\phi_S) = I_{\mathcal{S}}(\phi_S) \tag{$\lightning$}$$

does *not* hold. This is because in Abelian semirings idempotence does not necessarily hold, either. Also note that *De Morgan's laws* and *involution* ($\neg(\neg\phi_S) \equiv_{\mathcal{S}} \phi_S$) can be applied to (sub)sentences without structure elements only which, in the case of involution, means that ϕ_S must not mention structure elements. Otherwise, structure elements would be in the scope of negations which is not allowed in $\mathcal{SL}^S(\Sigma)$.

Other calculation rules hold under additional assumptions. One of these assumptions is the *mutually exclusiveness* of structured sentences.

Definition 6.2.13: Mutually Exclusive Structured Sentences
(cf. [Wilhelm et al., 2017a])

Let Σ be a finite signature, and let $\phi_S, \psi_S \in \mathcal{SL}^S(\Sigma)$ be structured sentences. If for all $I_{\mathcal{S}} \in \mathsf{Int}_{\mathcal{S}}(\Sigma)$ it holds that $I_{\mathcal{S}}(\phi_S) \neq 0$ implies $I_{\mathcal{S}}(\psi_S) = 0$, then ϕ_S and ψ_S are called *mutually exclusive*.

The mutually exclusiveness of structured sentences $\phi_S, \psi_S \in \mathcal{SL}^S(\Sigma)$ affects the interpretation of the disjunction $\phi_S \vee \psi_S$ in that the case "$I_{\mathcal{S}}(\phi_S \vee \psi_S) = I_{\mathcal{S}}(\phi_S) \otimes I_{\mathcal{S}}(\psi_S)$" (cf. Definition 6.2.4) cannot apply because $I_{\mathcal{S}}(\phi_S) = 0$ or $I_{\mathcal{S}}(\psi_S) = 0$ holds then. We prove that a weaker version of *distributivity* holds in $\mathcal{SL}^S(\Sigma)$ which presumes the mutually exclusiveness of structured sentences. This *weak distributivity* is of importance for some typed model counting techniques.

Proposition 6.2.14: Weak Distributivity

Let Σ be a finite signature, and let $\phi_S, \psi_S, \chi_S \in \mathcal{SL}^S(\Sigma)$ be structured sentences. If ϕ_S and ψ_S are mutually exclusive or if $I_{\mathcal{S}}(\chi_S)$ is idempotent for all $I_{\mathcal{S}} \in \mathsf{Int}_{\mathcal{S}}(\Sigma)$, e.g., if $\chi_S \in \mathcal{RL}^S(\Sigma)$, then

$$(\phi_S \vee \psi_S) \wedge \chi_S \equiv_{\mathcal{S}} (\phi_S \wedge \chi_S) \vee (\psi_S \wedge \chi_S). \tag{6.13}$$

We refer to the rule (6.13) as *weak distributivity.*

Proof. Let $I_{\mathcal{S}} \in \mathsf{Int}_{\mathcal{S}}(\Sigma)$ be a structured interpretation with $I_{\mathcal{S}}(\phi_S) \neq 0$. Because of the symmetry between ϕ_S and ψ_S in (6.13), the case in which $I_{\mathcal{S}}(\psi_S) \neq 0$ holds can be proven analogously. By assumption, $I_{\mathcal{S}}(\psi_S) = 0$, and, thus,

$$I_{\mathcal{S}}((\phi_S \vee \psi_S) \wedge \chi_S) = I_{\mathcal{S}}(\phi_S \vee \psi_S) \otimes I_{\mathcal{S}}(\chi_S) = I_{\mathcal{S}}(\phi_S) \otimes I_{\mathcal{S}}(\chi_S).$$

Note that $I_{\mathcal{S}}(\phi_S \vee \psi_S) = I_{\mathcal{S}}(\phi_S)$ holds because of $I_{\mathcal{S}}(\psi_S) = 0$. Likewise,

$$I_{\mathcal{S}}((\phi_S \wedge \chi_S) \vee I_{\mathcal{S}}(\psi_S \wedge \chi_S) = I_{\mathcal{S}}(\phi_S \wedge \chi_S) = I_{\mathcal{S}}(\phi_S) \otimes I_{\mathcal{S}}(\chi_S)$$

because

$$I_{\mathcal{S}}(\psi_S \wedge C) = I_{\mathcal{S}}(\psi_S) \otimes I_{\mathcal{S}}(\chi_S) = 0 \otimes I_{\mathcal{S}}(\chi_S) = 0.$$

If $I_{\mathcal{S}}(\phi_S) = I_{\mathcal{S}}(\psi_S) = 0$, then

$$I_{\mathcal{S}}((\phi_S \vee \psi_S) \wedge \chi_S) \equiv_{\mathcal{S}} I_{\mathcal{S}}((\phi_S \wedge \chi_S) \vee (\psi_S \wedge \chi_S)) \Leftrightarrow 0 \equiv_{\mathcal{S}} 0.$$

Eventually, we consider the case in which $I_{\mathcal{S}}(\chi_S)$ is idempotent. If $I_{\mathcal{S}}(\chi_S) = 0$, then

$$I_{\mathcal{S}}((\phi_S \vee \psi_S) \wedge \chi_S) \equiv_{\mathcal{S}} I_{\mathcal{S}}((\phi_S \wedge \chi_S) \vee (\psi_S \wedge \chi_S)) \Leftrightarrow 0 \equiv_{\mathcal{S}} 0.$$

Otherwise, if $I_{\mathcal{S}}(\chi_S) \neq 0$, then

$$\begin{aligned}
&I_{\mathcal{S}}((\phi_S \wedge \chi_S) \vee (\psi_S \wedge \chi_S)) \\
&= \begin{cases} I_{\mathcal{S}}(\phi_S \wedge \chi_S) & \text{if} \quad I_{\mathcal{S}}(\psi_S \wedge \chi_S) = 0 \\ I_{\mathcal{S}}(\psi_S \wedge \chi_S) & \text{if} \quad I_{\mathcal{S}}(\phi_S \wedge \chi_S) = 0 \\ I_{\mathcal{S}}(\phi_S \wedge \chi_S) \otimes I_{\mathcal{S}}(\psi_S \wedge \chi_S) & \text{otherwise} \end{cases} \\
&= \begin{cases} I_{\mathcal{S}}(\phi_S) \otimes I_{\mathcal{S}}(\chi_S) & \text{if} \quad I_{\mathcal{S}}(\psi_S) = 0 \\ I_{\mathcal{S}}(\psi_S) \otimes I_{\mathcal{S}}(\chi_S) & \text{if} \quad I_{\mathcal{S}}(\phi_S) = 0 \\ I_{\mathcal{S}}(\phi_S) \otimes I_{\mathcal{S}}(\chi_S) \otimes I_{\mathcal{S}}(\psi_S) \otimes I_{\mathcal{S}}(\chi_S) & \text{if} \quad I_{\mathcal{S}}(\phi_S) \neq 0 \neq I_{\mathcal{S}}(\psi_S) \end{cases} \\
&\overset{(*)}{=} \begin{cases} I_{\mathcal{S}}(\phi_S \vee \psi_S) \otimes I_{\mathcal{S}}(\chi_S) & \text{if} \quad I_{\mathcal{S}}(\psi_S) = 0 \\ I_{\mathcal{S}}(\psi_S \vee \phi_S) \otimes I_{\mathcal{S}}(\chi_S) & \text{if} \quad I_{\mathcal{S}}(\phi_S) = 0 \\ I_{\mathcal{S}}(\phi_S) \otimes I_{\mathcal{S}}(\psi_S) \otimes I_{\mathcal{S}}(\chi_S) & \text{if} \quad I_{\mathcal{S}}(\phi_S) \neq 0 \neq I_{\mathcal{S}}(\psi_S) \end{cases} \\
&= \begin{cases} I_{\mathcal{S}}((\phi_S \vee \psi_S) \wedge \chi_S) & \text{if} \quad I_{\mathcal{S}}(\psi_S) = 0 \\ I_{\mathcal{S}}((\psi_S \vee \phi_S) \wedge \chi_S) & \text{if} \quad I_{\mathcal{S}}(\phi_S) = 0 \\ I_{\mathcal{S}}(\phi_S \vee \psi_S) \otimes I_{\mathcal{S}}(\chi_S) & \text{if} \quad I_{\mathcal{S}}(\phi_S) \neq 0 \neq I_{\mathcal{S}}(\psi_S) \end{cases} \\
&= I_{\mathcal{S}}((\phi_S \vee \psi_S) \wedge \chi_S).
\end{aligned}$$

In the third case of the equality $(*)$, the idempotence of $I_{\mathcal{S}}(\chi_S)$ is applied. In the first two cases of $(*)$, it is used that $I_{\mathcal{S}}(\psi_S) = 0$ and $I_{\mathcal{S}}(\phi_S) = 0$ hold, respectively. ☐

The calculation rules in Proposition 6.2.12 and Proposition 6.2.14 are important for compiling structured sentences into specific normal forms that are well-suited for typed model counting. In the following, we define with $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$ a structured variant of the $\mathsf{sd\text{-}DNNF}$-normal form that is tailor-made for structured sentences.

▶ Structured Normal Form $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$

The *structured smooth deterministic decomposable negation normal form* $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$ is a normal form for structured sentences from $\mathcal{SL}^S(\Sigma)$ which is based on the $\mathsf{sd\text{-}DNNF}$-normal form for classical sentences (cf. Definition 6.1.2). Analogously to the requirement that the classical sentences in Definition 6.1.2 are quantifier-free, we assume in the definition of structured sentences in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$ that the structured sentences are free of quantifications. If this is not the case, the structured sentences can be made quantifier-free by the use of the equivalent conjunctions and disjunctions with domain constraints from (6.7) beforehand. The only reason for this preprocessing is that we want to avoid to discuss the handling of variables within $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal forms to make the definition of the $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal form a bit more concise. Note that, analogously to the classical case, $\mathsf{grAtom}(\phi_S)$ shall denote the set of ground atoms over Σ which occur in the structured sentence ϕ_S.

Definition 6.2.15: $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-Normal Form *(cf. [Wilhelm et al., 2019b])*

Let Σ be a finite signature, and let $\phi_S \in \mathcal{SL}^S(\Sigma)$ be a quantifier-free structured sentence (the equivalent conjunctions and disjunctions with domain constraints are allowed). Then, ϕ_S is in *structured negation normal form* ($\mathsf{NNF}^{\mathcal{S}}$) if negations in ϕ_S occur directly in front of ground atoms only. In addition:

- ϕ_S is *decomposable* iff every conjunction $\phi_{S_1} \wedge \ldots \wedge \phi_{S_m}$ in ϕ_S is *decomposable*, which means that $\mathsf{grAtom}(\phi_{S_i}) \cap \mathsf{grAtom}(\phi_{S_j}) = \emptyset$ for $i, j \in \{1, \ldots, m\}$ with $i \neq j$ holds.
- ϕ_S is *deterministic* iff every disjunction $\phi_{S_1} \vee \ldots \vee \phi_{S_m}$ in ϕ_S is *deterministic*, i.e., ϕ_{S_i} and ϕ_{S_j} are mutually exclusive for $i, j \in \{1, \ldots, m\}$ with $i \neq j$ (cf. Definition 6.2.13).
- ϕ_S is *smooth* iff both every disjunction $\phi_{S_1} \vee \ldots \vee \phi_m$ in ϕ_S is *smooth*, which means that $\mathsf{grAtom}(\phi_{S_i}) = \mathsf{grAtom}(\phi_{S_j})$ for $i, j \in \{1, \ldots, m\}$ holds, and if $\mathsf{grAtom}(\phi_S) = \mathsf{grAtom}(\Sigma)$ holds. The latter guarantees that all ground atoms from the underlying language are mentioned in ϕ_S. This condition can be dropped if the background language that is induced by the sentence ϕ_S is the whole structured language $\mathcal{SL}(\Sigma)$.

With $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}(\Sigma)$ we denote the set of all sentences $\phi_S \in \mathcal{SL}^S(\Sigma)$ which are in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal form, i.e., ϕ_S is in negation normal form, all conjunctions in ϕ_S are decomposable, and all disjunctions in ϕ_S are smooth and deterministic.

The decomposability of the conjunctions as well as the determinism of the disjunctions in structured sentences in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal form ensure that typed model counting can be modularized. This means that for sentences of the form $\phi_S = \phi_{S1} \wedge^d \ldots \wedge^d \phi_{Sm}$ it holds that

$$\mathsf{TMC}_{\Sigma(\phi_S)}(\phi_S) = \bigotimes_{i=1}^{m} \mathsf{TMC}_{\Sigma(\phi_{Si})}(\phi_{Si}),$$

and for sentences $\phi_S = \phi_{S1} \vee^{sd} \ldots \vee^{sd} \phi_S{}^m$ it holds that

$$\mathsf{TMC}_{\Sigma(\phi_S)}(\phi_S) = \bigoplus_{i=1}^{m} \mathsf{TMC}_{\Sigma(\phi_{Si})}(\phi_{Si}),$$

which is a consequence of the following Proposition 6.2.18.

Every structured sentence in $\mathcal{SL}^S(\Sigma)$ which is *satisfiable*, i.e., which has at least one typed model, can be compiled into $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal form. Otherwise, if $\phi_S \in \mathcal{SL}^S(\Sigma)$ is not satisfiable, then $\phi_S \equiv_{\mathcal{S}} \bot$.

Proposition 6.2.16: Compilation to $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-Normal Form

Let Σ be a finite signature, and let $\phi_S \in \mathcal{SL}^S(\Sigma)$ be a structured sentence which is satisfiable. Then, there is a structured sentence $\phi'_S \in \mathsf{sd\text{-}DNNF}^{\mathcal{S}}(\Sigma)$ which is in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal form such that $\phi_S \equiv_{\mathcal{S}} \phi'_S$ holds. Otherwise, if ϕ_S is not satisfiable, then $\phi_S \equiv_{\mathcal{S}} \bot$ holds.

Proof. Let ϕ_S be satisfiable. We give a straightforward proof of the proposition and consider the structured sentence

$$\phi'_S = \bigvee\nolimits^{sd}_{I^\omega_S \in \mathsf{Mod}^{\mathcal{S}}_{\Sigma}(\phi_S)\colon I^\omega_S \models \phi_S} I^\omega_S(\phi_S) \circ \omega.$$

The structured sentence ϕ'_S is in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$ because every two distinct possible worlds are mutually exclusive and mention the same ground atoms. Hence, the disjunction in ϕ'_S is smooth and deterministic. In addition, every possible world $\omega \in \Omega(\Sigma)$ is in $\mathsf{sd\text{-}DNNF}$-normal form because the conjunctions in ω are decomposable and, if so, negations occur in front of ground atoms.

By construction, ϕ'_S is $\mathcal{S}$-equivalent to ϕ_S. Let I^ω_S be an arbitrary structured interpretation in $\mathsf{Int}_S(\Sigma)$. The only disjunct in the disjunction in ϕ'_S which is satisfied in I^ω_S is $I^\omega_S(\phi_S) \circ \omega$ because $I^\omega_S(\omega') = 0$ holds for all $\omega' \in \Omega(\Sigma)$ with $\omega' \neq \omega$. Further,

$$I^\omega_S(\phi'_S) = I^\omega_S(I^\omega_S(\phi_S) \circ \omega) = I^\omega_S(\phi_S) \otimes I^\omega_S(\omega) = I^\omega_S(\phi_S) \otimes 1 = I^\omega_S(\phi_S),$$

such that $\phi_S \equiv_{\mathcal{S}} \phi'_S$ holds. If ϕ_S is not satisfiable, then $I_S(\phi_S) = 0$ for all structured interpretations $I_S \in \mathsf{Int}_S(\Sigma)$ which also holds for $\bot$ such that $\phi_S \equiv_S \bot$ follows. Note that ϕ'_S proves the existence of a structured sentence which is $\mathcal{S}$-equivalent to ϕ_S but does not allow for tractable typed model counting. For the latter purpose, more compact $\mathcal{S}$-equivalent structured sentences have to be found. Strategies for that are discussed later on. $\square$

The counting of the typed models of a structured sentence in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$ works similar to the counting of the models of sentences in $\mathsf{sd\text{-}DNNF}$ apart from the fact that structured sentences are mapped to elements from the semiring $\mathcal{S}$ instead of natural numbers. For the formal counting, we define a structured pendant to the count function in Definition 6.1.5.

Definition 6.2.17: Structured Count Function
Let Σ be a finite signature. We recursively define the *structured count function* $\mathsf{cnt}_{\mathcal{S}} \colon \mathsf{sd\text{-}DNNF}^{\mathcal{S}}(\Sigma) \to \mathcal{S}$ which assigns to every structured sentence in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal form its typed model count by $\mathsf{cnt}_{\mathcal{S}}(\top) = 1$, $\mathsf{cnt}_{\mathcal{S}}(\bot) = 0$, $\mathsf{cnt}_{\mathcal{S}}(l) = 1$ for $l \in \mathsf{grLit}(\Sigma)$, and $\mathsf{cnt}_{\mathcal{S}}(s \circ \phi_S) = s \otimes \mathsf{cnt}_{\mathcal{S}}(\phi_S)$ for $s \in \mathcal{S}$ and $\phi_S \in \mathsf{sd\text{-}DNNF}^{\mathcal{S}}(\Sigma)$. Further, for every decomposable conjunction and every smooth deterministic disjunction, we have

$$\begin{aligned}\mathsf{cnt}_{\mathcal{S}}(\phi_{S1} \wedge^{d} \ldots \wedge^{d} \phi_{Sm}) &= \bigotimes_{i=1}^{m} \mathsf{cnt}_{\mathcal{S}}(\phi_{Si}),\\ \mathsf{cnt}_{\mathcal{S}}(\phi_{S1} \vee^{sd} \ldots \vee^{sd} \phi_{Sm}) &= \bigoplus_{i=1}^{m} \mathsf{cnt}_{\mathcal{S}}(\phi_{Si}).\end{aligned}$$

Note that the structured formulas $\phi_{S1}, \ldots, \phi_{Sm}$ in Definition 6.2.17 are structured sentences in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$. Based on the structured count function, we have the following proposition.

Proposition 6.2.18: Correctness of Structured Count Function
Let Σ be a finite signature, and let $\phi_S \in \mathsf{sd\text{-}DNNF}^{\mathcal{S}}(\Sigma)$ be a structured sentence in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal form. Then,

$$\mathsf{cnt}_{\mathcal{S}}(\phi_S) = \mathsf{TMC}_{\Sigma}(\phi_S). \tag{6.14}$$

Proof. We prove the proposition by structural induction. Let $\Sigma(\psi_S)$ denote the signature of the structured sentence ψ_S, and let $l \in \mathsf{grLit}(\Sigma)$ be a ground literal. Then, obviously

$$\mathsf{cnt}_{\mathcal{S}}(l) = 1 = \mathsf{TMC}_{\Sigma(l)}(l).$$

In addition, the base cases $\mathsf{cnt}_{\mathcal{S}}(\top) = 1 = \mathsf{TMC}_{\emptyset}(\top)$ and $\mathsf{cnt}_{\mathcal{S}}(\bot) = 0 = \mathsf{TMC}_{\emptyset}(\bot)$ hold trivially. Now, let $s \in \mathcal{S}$ and $\phi_S \in \mathsf{sd\text{-}DNNF}^{\mathcal{S}}(\Sigma)$. Then,

$$\mathsf{cnt}_{\mathcal{S}}(s \circ \phi_S) = s \otimes \mathsf{cnt}_{\mathcal{S}}(\phi_S) = s \otimes \mathsf{TMC}_{\Sigma(\phi_S)}(\phi_S) = \mathsf{TMC}_{\Sigma(\phi_S)}(s \circ \phi_S).$$

Further, let $\phi_S = \phi_{S1} \vee^{sd} \ldots \vee^{sd} \phi_{Sm}$ be a smooth and deterministic disjunction. Because the disjunction is smooth, all disjuncts have the same signature, say Σ'. We have

$$\begin{aligned}\mathsf{cnt}_{\mathcal{S}}(\phi_{S1} \vee^{sd} \ldots \vee^{sd} \phi_{Sm}) = \bigoplus_{i=1}^{m} \mathsf{cnt}_{\mathcal{S}}(\phi_{Si}) = \bigoplus_{i=1}^{m} \mathsf{TMC}_{\Sigma'}(\phi_{Si})& \\ = \mathsf{TMC}_{\Sigma'}(\phi_{S1} \vee^{sd} \ldots \vee^{sd} \phi_{Sm}).&\end{aligned}$$

The last equality holds because of the determinism of the conjunctions which prevents us from counting one and the same typed model twice.

Eventually, let $\phi_S = \phi_{S1} \wedge^{d} \ldots \wedge^{d} \phi_{Sm}$ be a decomposable conjunction with $\Sigma' = \Sigma(\phi_S)$. Then,

$$\begin{aligned}\mathsf{cnt}_{\mathcal{S}}(\phi_{S1} \wedge^{d} \ldots \wedge^{d} \phi_{Sm}) = \bigotimes_{i=1}^{m} \mathsf{cnt}_{\mathcal{S}}(\phi_{Si}) = \bigotimes_{i=1}^{m} \mathsf{TMC}_{\Sigma(\phi_{Si})}(\phi_{Si})& \\ = \mathsf{TMC}_{\Sigma'}(\phi_{S1} \wedge^{d} \ldots \wedge^{d} \phi_{Sm}).&\end{aligned}$$

Here, the last equality holds because all signatures $\Sigma(\phi_{Si})$ for $i = 1, \ldots, m$ are pairwise disjoint and the question whether a structured interpretation in $\mathsf{Int}_{\mathcal{S}}(\Sigma(\phi_{Si}))$ is a typed model of ϕ_{Si} is not affected by the evaluation of ϕ_{Sj} for $j \neq i$. $\square$

Note that the signature Σ is not explicitly mentioned on the left-hand side of (6.14) in Proposition 6.2.18, i.e., in $\mathsf{cnt}_{\mathcal{S}}(\phi_{\mathcal{S}})$, while the right-hand side, $\mathsf{TMC}_{\Sigma}(\phi_{\mathcal{S}})$, explicitly mentions Σ. This is because for all sentences ϕ_S in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$ we have $\mathsf{grAtom}(\phi_S) = \mathsf{grAtom}(\Sigma)$ because of the smoothness property of ϕ_S. Hence, the signature Σ is implicitly determined by ϕ_S.

Example 6.2.19

The structured sentence

$$\psi_S = (s_1 \circ C \vee \overline{C}) \wedge (s_2 \circ (M \wedge P) \vee ((\overline{M} \wedge (P \vee \overline{P})) \vee (M \wedge \overline{P})))$$

from Example 6.2.11 is in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal form. The typed models of ψ_S can be computed using the structured count function as follows[a]:

$$\begin{aligned}&\mathsf{cnt}_{\mathcal{S}}(\psi_S)\\ &= \mathsf{cnt}_{\mathcal{S}}((s_1 \circ C \vee \overline{C}) \wedge (s_2 \circ (M \wedge P) \vee ((\overline{M} \wedge (P \vee \overline{P})) \vee (M \wedge \overline{P}))))\\ &= \mathsf{cnt}_{\mathcal{S}}(s_1 \circ C \vee \overline{C}) \cdot \mathsf{cnt}_{\mathcal{S}}((s_2 \circ (M \wedge P) \vee ((\overline{M} \wedge (P \vee \overline{P})) \vee (M \wedge \overline{P}))))\\ &= (\mathsf{cnt}_{\mathcal{S}}(s_1 \circ C) + \mathsf{cnt}_{\mathcal{S}}(\overline{C})) \cdot \mathsf{cnt}_{\mathcal{S}}(s_2 \circ (M \wedge P))\\ &\qquad + \mathsf{cnt}_{\mathcal{S}}((\overline{M} \wedge (P \vee \overline{P})) \vee (M \wedge \overline{P}))\end{aligned}$$

$$\begin{aligned}
&= (s_1 \cdot \mathsf{cnt}_{\mathcal{S}}(C)) + \mathsf{cnt}_{\mathcal{S}}(\overline{C})) \cdot (s_2 \cdot \mathsf{cnt}_{\mathcal{S}}(M \wedge P) + \mathsf{cnt}_{\mathcal{S}}(\overline{M} \wedge (P \vee \overline{P})) \\
&\qquad + \mathsf{cnt}_{\mathcal{S}}(M \wedge \overline{P})) \\
&= (s_1 \cdot \mathsf{cnt}_{\mathcal{S}}(C)) + \mathsf{cnt}_{\mathcal{S}}(\overline{C})) \cdot (s_2 \cdot \mathsf{cnt}_{\mathcal{S}}(M) \cdot \mathsf{cnt}_{\mathcal{S}}(P) + \mathsf{cnt}_{\mathcal{S}}(\overline{M}) \\
&\qquad \cdot (\mathsf{cnt}_{\mathcal{S}}(P) + \mathsf{cnt}_{\mathcal{S}}(\overline{P})) + \mathsf{cnt}_{\mathcal{S}}(M) \cdot \mathsf{cnt}_{\mathcal{S}}(\overline{P})) \\
&= ((s_1 \cdot 1) + 1) \cdot (s_2 \cdot 1 \cdot 1 + 1 \cdot (1 + 1) + 1 \cdot 1) \\
&= (s_1 + 1) \cdot (s_2 + 2 + 1) \\
&= (s_1 + 1) \cdot (s_2 + 3) \qquad (*) \\
&= s_1 s_2 + 3s_1 + s_2 + 3 \\
&= \mathsf{TMC}_\Sigma(\psi_S).
\end{aligned}$$

Recall that ψ_S is equivalent to ϕ_S from Example 6.2.1, so you may confer $\mathsf{TMC}_\Sigma(\psi_S)$ with Table 6.1. The product representation $(*)$ of $\mathsf{TMC}_\Sigma(\psi_S)$ nicely shows the influence of the two cases "being a computer scientist" and "being a mathematician with a passion for computer science" on $\mathsf{TMC}_\Sigma(\psi_S)$. Thereby, $(s_1 + 1)$ refers to "being a computer scientist or not" and $(s_2 + 3)$ to "being a mathematician with a passion for computer science or not". Because in the latter case two conditions have to come together—"being a mathematician" and "having a passion for computer science"—, there are three subcases in which "being a mathematician with a passion for computer science" does not hold, hence, in which the typed model of the respective subsentence is 1 instead of s_2.

---

[a] Recall that we have defined the semiring $\mathcal{S}$ by $\mathcal{S} = (\{s_1, s_2\}, +, \cdot, 0, 1)$ in Example 6.2.3, i.e., we use the common symbols $\cdot$ and $+$ for multiplication and addition, respectively, here.

For conjunctions and disjunctions with domain constraints, the definition of the structured count function results in

$$\mathsf{cnt}_{\mathcal{S}}\Big(\bigwedge\nolimits^{d}_{c \in \mathsf{Sol}(\mathsf{CS})} \phi_S\langle X/c\rangle\Big) = \bigotimes_{c \in \mathsf{Sol}(\mathsf{CS})} \mathsf{cnt}_{\mathcal{S}}(\phi_S\langle X/c\rangle),$$
$$\mathsf{cnt}_{\mathcal{S}}\Big(\bigvee\nolimits^{d}_{c \in \mathsf{Sol}(\mathsf{CS})} \phi_S\langle X/c\rangle\Big) = \bigoplus_{c \in \mathsf{Sol}(\mathsf{CS})} \mathsf{cnt}_{\mathcal{S}}(\phi_S\langle X/c\rangle).$$

If for all $c_1, c_2 \in \mathsf{Sol}(\mathsf{CS})$ it holds that $\mathsf{cnt}_{\mathcal{S}}(\phi_S\langle X/c_1\rangle) = \mathsf{cnt}_{\mathcal{S}}(\phi_S\langle X/c_2\rangle)$, in addition, then the total count of the conjunction respectively the disjunction simplifies to

$$\mathsf{cnt}_{\mathcal{S}}\Big(\bigwedge\nolimits^{d}_{c \in \mathsf{Sol}(\mathsf{CS})} \phi_S\langle X/c\rangle\Big) = \mathsf{cnt}_{\mathcal{S}}(\phi_S\langle X/c_1\rangle)^{|\mathsf{Sol}(\mathsf{CS})|},$$
$$\mathsf{cnt}_{\mathcal{S}}\Big(\bigvee\nolimits^{d}_{c \in \mathsf{Sol}(\mathsf{CS})} \phi_S\langle X/c\rangle\Big) = |\mathsf{Sol}(\mathsf{CS})| \cdot \mathsf{cnt}_{\mathcal{S}}(\phi_S\langle X/c_1\rangle).$$

Like in the unstructured case, the condition $\mathsf{cnt}_{\mathcal{S}}(\phi_S\langle X/c_1\rangle) = \mathsf{cnt}_{\mathcal{S}}(\phi_S\langle X/c_2\rangle)$

holds if $\phi_S\langle X/c_1\rangle$ and $\phi_S\langle X/c_2\rangle$ are *isomorphic*, i.e., if $\phi_S\langle X/c_1\rangle$ and $\phi_S\langle X/c_2\rangle$ are identical up to a renaming of constants.

We define a circuit-representation for structured sentences in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$ in analogy to circuits for classical sentences in $\mathsf{sd\text{-}DNNF}$.

Definition 6.2.20: $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-Circuit *(cf. Wilhelm et al. [2017a])*

Let Σ be a finite signature, and let $\phi_S \in \mathsf{sd\text{-}DNNF}^{\mathcal{S}}(\Sigma)$ be a structured sentence in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal form. Then, we recursively define the $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-*circuit* $\mathsf{circuit}(\phi_S)$ of ϕ_S as a labeled directed acyclic graph as follows:

- If ϕ_S is of the form $\top$ (or $\bot$), then $\mathsf{circuit}(\phi_S)$ is a single node with label ϕ_S.
- If $\phi_S \in \mathsf{grLit}(\Sigma)$, then $\mathsf{circuit}(\phi_S)$ is a single node with label ϕ_S.
- If ϕ_S is of the form $s \circ \psi_S$ with $s \in \mathcal{S}_{\mathfrak{G}}$, then $\mathsf{circuit}(\phi_S)$ is a node labeled with s and with $\mathsf{circuit}(\psi_S)$ as a child graph.[a]
- If ϕ_S is a conjunction $\phi_{S_1} \wedge \ldots \wedge \phi_{S_m}$, then $\mathsf{circuit}(\phi_S)$ is a node with label $\wedge$ and with child graphs $\mathsf{circuit}(\phi_{S_i})$ for $i = 1, \ldots, m$.
- If ϕ_S is a disjunction $\phi_{S_1} \vee \ldots \vee \phi_{S_m}$, then $\mathsf{circuit}(\phi_S)$ is a node with label $\vee$ and with child graphs $\mathsf{circuit}(\phi_{S_i})$ for $i = 1, \ldots, m$.
- If ϕ_S is a conjunction with domain constraints $\bigwedge_{c\in\mathsf{Sol}(\mathsf{CS})} \psi_S\langle X/c\rangle$, then $\mathsf{circuit}(\phi_S)$ is a node with label $\bigwedge_{c\in\mathsf{Sol}(\mathsf{CS})}$ and with child graph $\mathsf{circuit}(\psi_S\langle X/c\rangle)$.
- If ϕ_S is a disjunction with domain constraints $\bigvee_{c\in\mathsf{Sol}(\mathsf{CS})} \psi_S\langle X/c\rangle$, then $\mathsf{circuit}(\phi_S)$ is a node with label $\bigvee_{c\in\mathsf{Sol}(\mathsf{CS})}$ and with child graph $\mathsf{circuit}(\psi_S\langle X/c\rangle)$.

---

[a] We say that a directed acyclic graph G is a *child graph* of a node N if there is a directed edge from N to the root node of G.

$\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-circuits are defined analogously to $\mathsf{sd\text{-}DNNF}$-circuits except for the additional inner nodes which mention the structure elements from $\mathcal{S}_{\mathfrak{G}}$.

Example 6.2.21

The $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-circuit of the structured sentence ψ_S from Example 6.2.24 is shown in Figure 6.3.

We also define a structured pendant to counting graphs.

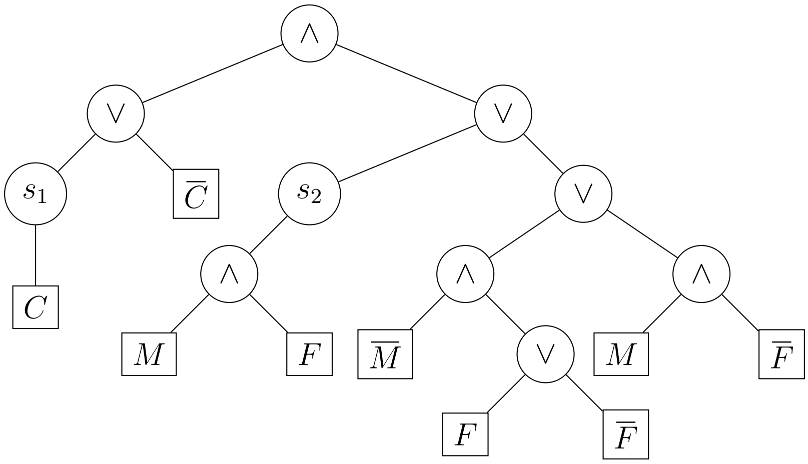

Figure 6.3: $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-Circuit of the sentence ψ_S from Example 6.2.24.

Definition 6.2.22: Structured Counting Graph

(cf. [Wilhelm et al., 2017a])

Let Σ be a finite signature, and let $\phi_S \in \mathsf{sd\text{-}DNNF}^{\mathcal{S}}(\Sigma)$ be a structured sentence in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal form. The *structured counting graph* of ϕ_S is obtained from the $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-circuit $\mathsf{circuit}(\phi_S)$ by making the following replacements:

- Each node label which is $\top$ (or $\bot$) is replaced by 1 (or 0).
- Each node label which is a ground literal from $\mathsf{grLit}(\Sigma)$ is replaced by 1.
- Each node label $\wedge$ is replaced by $\otimes$.
- Each node label $\vee$ is replaced by $\oplus$.
- Each node label of the form $\bigwedge_{c\in\mathsf{Sol(CS)}}$ is replaced by $(\ldots)^{|\mathsf{Sol(CS)}|}$.
- Each node label of the form $\bigvee_{c\in\mathsf{Sol(CS)}}$ is replaced by $|\mathsf{Sol(CS)}|\cdot(\ldots)$.

The third graphical representation that we use for structured sentence ϕ_S in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal form is the $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$*-circuit with counts* which is the $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-circuit of ϕ_S with the cumulated model counts annotated next to the nodes. In particular, in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-circuits with counts, the typed model count of ϕ_S is written next to the root node of the $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-circuit of ϕ_S.

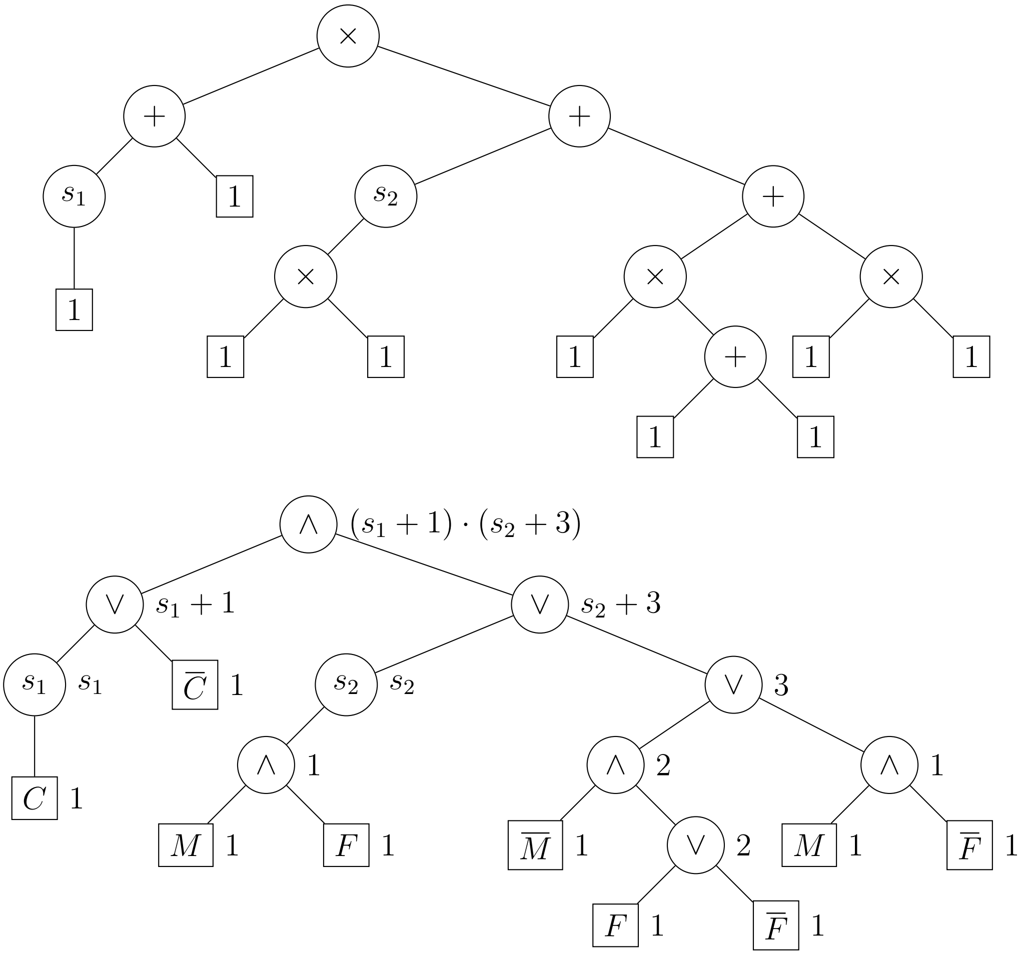

Figure 6.4: Structured counting graph (top) and sd-DNNF-circuit with counts (bottom) of the sentence T from Example 6.2.24.

Example 6.2.23
In Figure 6.4, the structured counting graph of the structured sentence ψ_S from Example 6.2.11 is shown. Also shown in Figure 6.4 is the sd-DNNF$^{\mathcal{S}}$-circuit with counts of ψ_S. See also Figure 6.3 for comparison.

We give another example which demonstrates how typed model counting can be performed on a structured sentence that involves a quantifier. Note that the resulting typed model count is parameterized by the domain size. Therewith, we can compute the typed model count with respect to any concrete domain simply by inserting the domain size into the expression. Also note that we make use of the concept of set constraints in this example (cf. Section 2.3).

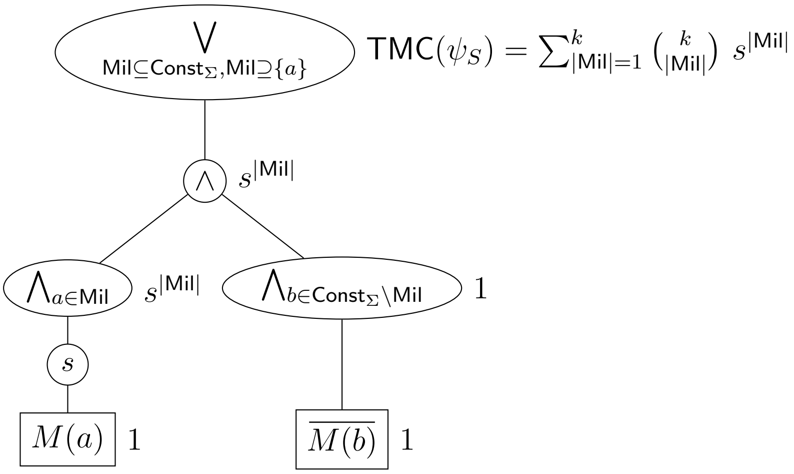

Figure 6.5: $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-circuit of the structured sentence ψ_S from Example 6.2.24.

Example 6.2.24
We consider the structured sentence

$$\psi_S = \exists X.s \circ M(X)$$

"There is at least one millionaire."

over the signature $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ with $\mathsf{Pred}_\Sigma = \{M/1\}$ and an arbitrary set of constants Const_Σ with $k = |\mathsf{Const}_\Sigma| \geq 2$. Further, s is the structure element in ψ_S. The single instances of the existential quantification in ψ_S are not mutually exclusive so that distinct ground atoms $M(a)$ and $M(b)$ may exist which both are mapped to 1 by the same structured interpretation. Informally, this means that there is a structured interpretation in which more than one millionaire exists. In order to compile ψ_S into an $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-circuit, it is expedient to distinguish the structured interpretations by the number of ground instances of $M(X)$ they map to 1—the number of millionaires—and to divide the set of constants into millionaires and non-millionaires. This can be achieved via set and domain constraints as shown in Figure 6.5. Note that $\binom{k}{|\mathsf{Mil}|}$ is the number of possibilities to select $|\mathsf{Mil}|$-many millionaires out of k individuals. The resulting typed model count

$$\mathsf{TMC}_\Sigma(\psi_S) = \sum_{|\mathsf{Mil}|=1}^{k} \binom{k}{|\mathsf{Mil}|}\, s^{|\mathsf{Mil}|}$$

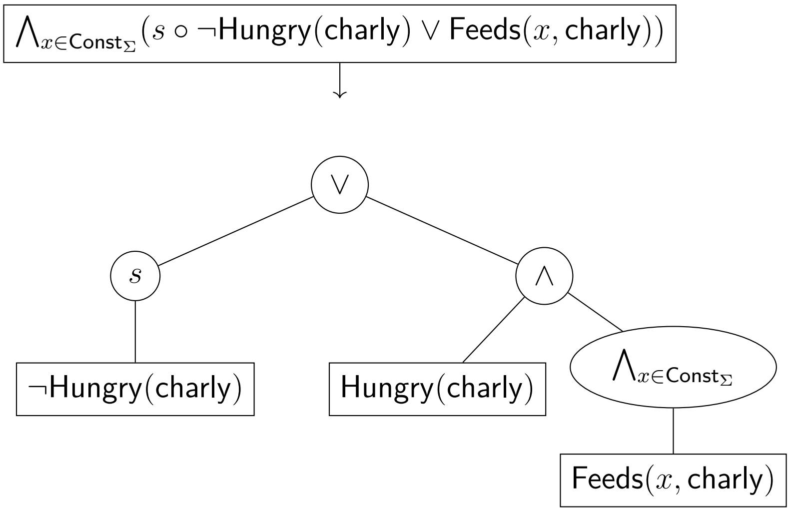

Figure 6.6: Illustration of the compilation technique *Shannon decomposition.* The decomposition is performed with respect to the ground atom Hungry(charly). Note that after applying the Shannon decomposition all conjunctions in this structured sentence are decomposable as well.

is parameterized by the domain size k. If we stipulate $k = 3$, we get

$$\mathsf{TMC}_\Sigma(\psi_S) = 3 \cdot s + 3 \cdot s^2 + s^3. \qquad (k = 3)$$

This expression can be interpreted as follows: Under all typed models of ψ_S, we can find three structured interpretations in which there is exactly one millionaire ($3 \cdot s$), three structured interpretations in which there are exactly two millionaires ($3 \cdot s^2$), and there is one structured interpretation in which there are three millionaires (s^3).

It is important to note that the $\mathsf{sd\text{-}DNNF}^\mathcal{S}$-normal form of a satisfiable structured sentence $\phi_S \in \mathcal{SL}^S(\Sigma)$ is not unique. In particular, the compactness of the compilation of ϕ_S into $\mathsf{sd\text{-}DNNF}^\mathcal{S}$-normal form depends on the applied compilation techniques. This means, we encounter the same questions as in classical knowledge compilation here (cf. [Darwiche and Marquis, 2002]), and it is a goal to lift knowledge compilation techniques from classical sentences to structured sentences. In the following, we name some important compilation techniques that are discussed in [Van den Broeck, 2013] and demonstrate how they could be lifted to structured sentences by means of examples:[1]

Shannon Decomposition: The idea of *Shannon decomposition* is to split a sentence into mutually exclusive cases with respect to the evaluation of an atom

---

[1] Applying these rules does not guarantee smoothness which possibly has to be recovered afterwards.

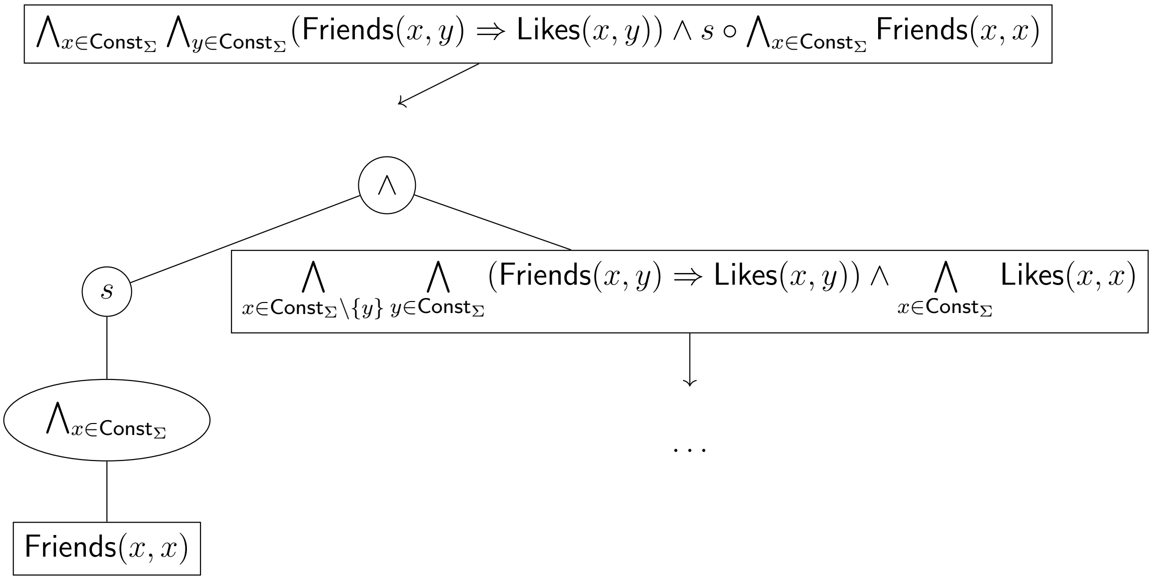

Figure 6.7: Illustration of the compilation technique *unit propagation*. Here, the conjunct stating that being friends is a reflexive relation is separated. Notice how this affects the scope of the quantifier in the other conjunct.

and its negation. A formalization of the Shannon decomposition for classical sentences and ground atoms can be found in (6.11). Please see Section 4.2.3 of [Van den Broeck, 2013] for more details. Figure 6.6 shows a possible adaption of the Shannon decomposition to structured sentences. With the Shannon decomposition it is possible to establish deterministic conjunctions.

Unit Propagation: The idea of *unit propagation* is to separate conjuncts that consist of single literals. After applying unit propagation, the literal which is separated does not occur in the remainder of the conjunction anymore so that the conjunction becomes decomposable. Details on the concept of unit propagation can be found in Section 4.2.1 of Van den Broeck [2013]. Figure 6.7 shows an adaption to structured sentences.

Shattered Compilation: The idea of *shattered compilation* is to isolate isomorphic expressions like in the example shown in Figure 6.8. Shattering is rather a precompilation step than a standalone compilation rule. In the example shown in Figure 6.8, the initial conjunctions are shattered into the cases where $x = y$ and where $x \neq y$. After that, unit propagation is applied to separate the symmetric part in which $x = y$ holds. More about shattered compilation can be found in Section 4.3.4 of [Van den Broeck, 2013].

Atom Counting: The idea of *atom counting* is to partition the domain with respect to the evaluation of an atom with a single variable. This technique refers

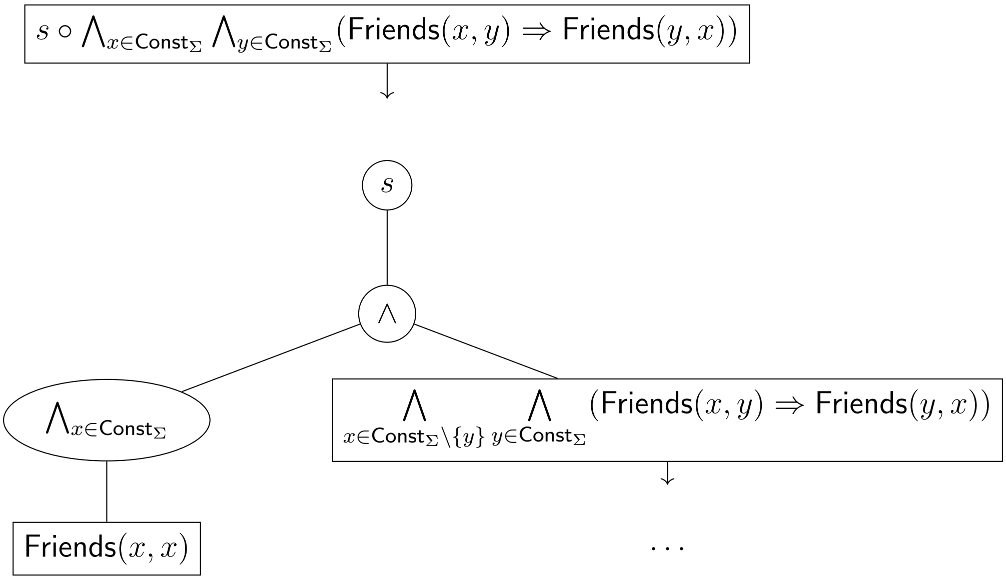

Figure 6.8: Illustration of the compilation technique *shattered compilation*. Here, the conjunct stating that being friends is a reflexive relation is separated. This conjunct cannot be isomorphic to the others because the first and the second argument of Friends/2 has to be instantiated with the same constant.

> to our approach of making use of set constraints in Example 6.2.24. In this example, the atom $M(X)$ ("X is a millionaire") occurred so that we split the domain into the millionaires, i.e., the individuals c which satisfy $M(c)$, and the non-millionaires, i.e., individuals c for which $\overline{M(c)}$ holds. This generated a deterministic disjunction over the set variable Mil. Please see Section 4.4.6 of [Van den Broeck, 2013] for further details.

If no other compilation technique is applicable, then it is always possible to proceed as in the proof of Proposition 6.2.16 which correlates to the act of *grounding*. However, note that it is our goal to avoid grounding whenever it is possible because grounding makes lifted reasoning impossible. For instance, the atom counting technique applied to ψ_S in Example 6.2.24 reduces the number of summands in $\mathsf{TMC}_\Sigma(\psi_S)$ from 2^{k-1} (when naïve grounding is applied) to k. Before we apply typed model counting to conditional knowledge bases in order to compute conditional structures, we compare typed model counting with the related algebraic model counting.

6.3 Typed Versus Algebraic Model Counting

Typed model counting (Section 6.2) and algebraic model counting (Section 6.1) are similar approaches in the sense that both approaches extend first-order model counting by the use of an Abelian semiring in order to enhance the interpretation of sentences. In this section, we work out differences between the two approaches as well as commonalities of them. We provide a translation of structured sentences to algebraically interpreted sentences and, therewith, of the typed model counting task to the algebraic model counting task. Finally, we note some advantages of typed model counting.

▶ Translation of Structured Sentences

An obvious difference between typed model counting and algebraic model counting is that in typed model counting algebraic elements are incorporated into logical sentences directly while algebraic model counting treats these elements externally. A more conceptual difference between typed model counting and algebraic model counting is that in algebraic model counting the elements from the semiring are associated to the ground atoms that are mentioned in the sentence, while in typed model counting the structure elements are associated to (sub)sentences. Nevertheless, it is possible to translate the typed model counting task into an algebraic model counting task. The main idea of this translation is to introduce for every structure element $s \in \mathcal{S}_{\mathfrak{S}}$ which occurs in a structured sentence $\phi_S \in \mathcal{SL}^S(\Sigma)$ a fresh ground atom A_s which is interpreted to s. However, the naïve idea of simply replacing every occurrence of s by A_s does not work.

A problem with the naïve translation of structured sentences from $\mathcal{SL}^S(\Sigma)$ into an algebraically interpreted sentence in $\mathcal{RL}^S(\Sigma)$ is that the fresh ground atoms enlarge the underlying signature and, hence, have an influence on both the number of the interpretations and also on the interpretations of sentences themselves. In general, the naïve translation of structured sentences into algebraically interpreted classical sentences does not preserve the model count of the sentence. We give an example which illustrates the problem with the naïve translation of structured sentences first, and discuss a correct translation afterwards.

Example 6.3.1
We consider the structured sentence

$$\phi_S = (s \circ A) \vee B$$

over the signature $\Sigma = (\emptyset, \{A/0, B/0\})$, where s is a structure element. The

ω	$I_S^\omega(\phi_S)$	ω	$I_S^\omega(\phi_S)$
AB	s	$\overline{A}B$	1
$A\overline{B}$	s	$\overline{A}\overline{B}$	0

ω	$I_\rho^\omega(\phi')$	ω	$I_\rho^\omega(\phi')$
ABA_s	s	$\overline{A}BA_s$	s
$AB\overline{A_s}$	1	$\overline{A}B\overline{A_s}$	1
$A\overline{B}A_s$	s	$\overline{A}\overline{B}A_s$	0
$A\overline{B}\overline{A_s}$	0	$\overline{A}\overline{B}\overline{A_s}$	0

Table 6.3: Structured interpretations of the structured sentence ϕ_S (top) and algebraic interpretations of the sentence ϕ' (bottom) from Example 6.3.1, where the algebraic interpretations are determined by the assignment ρ as stated in Example 6.3.1.

naïve translation of ϕ_S into an algebraically interpreted sentence is

$$\phi' = (A_s \wedge A) \vee B$$

where A_s is a fresh ground atom that is interpreted to s. More precisely, the algebraic interpretations of ϕ' are given by the assignment ρ with $\rho(A_s) = s$ and $\rho(l) = 1$ for all remaining ground literals l in the language (cf. (6.6) for the meaning of assignments). The structured interpretations of ϕ_S and the algebraic interpretations of ϕ' with respect to ρ are shown in Table 6.3. Not only the number of interpretations in total change when translating ϕ_S to ϕ'_S but also the number of models. In addition, the proportion of the different model types vary. In particular, we obtain the counterintuitive algebraic interpretation $I_\rho^\omega(\phi') = s$ for $\omega = \overline{A}BA_s$. In this point the algebraic interpretation of ϕ' differs from the structured interpretation of ϕ_S because A_s and A are not forced by ϕ' to be satisfied simultaneously, while s is a constituent of the type of a structured interpretation of ϕ_S if and only if A is true.

In order to find a proper translation of structured sentences to algebraically interpreted sentences, we have a closer look at the definition of structured sentences in Definition 6.2.2. Recall that structured sentences extend sentences from $\mathcal{RL}^S(\Sigma)$ by the possibility to concatenate a structure element to the left. Accordingly, the only construction rule in (6.12) in which structure elements come into play is the one which handles the case "$s \circ \psi_S$" where s is a structure element and ψ_S is a structured

ω	$I^{\omega}_{\rho}(\nu(\phi_S))$	ω	$I^{\omega}_{\rho}(\nu(\phi_S))$
ABA_s	s	$\overline{A}BA_s$	0
$AB\overline{A_s}$	0	$\overline{A}B\overline{A_s}$	1
$A\overline{B}A_s$	s	$\overline{A}\overline{B}A_s$	0
$A\overline{B}\overline{A_s}$	0	$\overline{A}\overline{B}\overline{A_s}$	0

Table 6.4: Algebraic interpretations of the sentence $\nu(\phi_S)$ from Example 6.3.2 with respect to the assignment ρ as stated in Example 6.3.2.

sentence. We analyze this case in detail now. Without loss of generality, we limit our investigations to structured sentences without quantifiers as we have already done for the definition of structured sentences in $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal form (Definition 6.2.15).

Let $\phi_S \in \mathcal{SL}^{S}(\Sigma)$ be a quantifier-free structured sentence, and let us assume—for the moment—that all subsentences of ϕ_S which are of the form $s \circ \psi_S$ with $s \in \mathcal{S}_{\mathfrak{S}}$ and $\psi_S \in \mathcal{SL}^{S}(\Sigma)$ do not mention a structure element within ψ_S. This is why we write ψ instead of ψ_S from now on. We will generalize this case later on by removing the restriction that there is no structure element in ψ but, with this restriction, the idea of translating ϕ_S into an algebraically interpreted sentence works as follows: Instead of replacing $s \circ \psi$ by $A_s \wedge \psi$ with respect to a fresh ground atom A_s directly, as done in the naïve translation, we replace $s \circ \psi$ by ψ, denoted by $\phi_S\langle s \circ \psi / \psi\rangle$, and append the conjunct $(A_s \wedge \psi) \vee (\overline{A_s} \wedge \overline{\psi})$ to ϕ_S. That is, we translate ϕ_S to

$$\phi'_S = \phi_S\langle s \circ \psi / \psi\rangle \wedge ((A_s \wedge \psi) \vee (\overline{A_s} \wedge \overline{\psi})). \tag{6.15}$$

The fact that ϕ_S does not mention variables ensures that ϕ'_S is a structured sentence free of variables again. The conjunct $(A_s \wedge \psi) \vee (\overline{A_s} \wedge \overline{\psi})$ guarantees that A_s is interpreted to s if and only if ψ is true. By a repeated application of the rewriting rule (6.15), the sentence ϕ_S can be freed from all structure elements.

In the general case it may happen that there is a subsentence $s \circ \psi_S$ of ϕ_S where ψ_S mentions a structure element itself, the case we excluded from our deliberations above. Then one applies the rewriting rule (6.15) to the structure elements in ψ_S first and, therewith, frees ψ_S from structure elements before one continues with $s \circ \psi_S$. That is, one begins with the innermost structure elements. We denote the translation of ϕ_S after removing all structure elements from ϕ_S by the repeated application of (6.15) with $\nu(\phi_S)$.

Example 6.3.2

We consider two examples.

a) First, we consider the structured sentence

$$\phi_S = (s \circ A) \vee B$$

from Example 6.3.1. The proper translation of ϕ_S is

$$\nu(\phi_S) = (A \vee B) \wedge ((A_s \wedge A) \vee (\overline{A_s} \wedge \overline{A})),$$

and the algebraic interpretation of $\nu(\phi_S)$ with respect to the assignment ρ such that $\rho(A_s) = s$ and $\rho(l) = 1$ for all remaining ground literals l is shown in Table 6.4. The number of algebraic interpretations of $\nu(\phi_S)$ still differs from the number of structured interpretations of ϕ_S from Example 6.3.1 because the number of ground atoms has been increased, but the algebraic model count of $\nu(\phi_S)$ equals the typed model count of ϕ_S now:

$$\mathsf{AFOMC}^{\rho}_{\Sigma'}(\nu(\phi_S)) = 2 \cdot s + 1 = \mathsf{TMC}_{\Sigma}(\phi_S),$$

where Σ' is Σ extended by A_s. In plain words, with the fresh ground atom A_s we increase the number of interpretations which do not satisfy the sentence but not the number of its models.

b) Second, we consider the structured sentence

$$\psi_S = s_1 \circ ((s_2 \circ A) \vee B)$$

with respect to $\Sigma = (\emptyset, \{A/0, B/0\})$ and the structured elements s_1 and s_2. Then, the translation of ψ_S to an equivalent algebraically interpreted sentence can be recursively performed as follows:

$$\begin{aligned}\nu(\psi_S) &= \nu(s_1 \circ (A \vee B)) \wedge (A_{s_2}A \vee \overline{A_{s_2}}\overline{A}) \\ &= (A \vee B) \wedge (A_{s_2}A \vee \overline{A_{s_2}}\overline{A}) \wedge (A_{s_1}(A \vee B) \vee \overline{A_{s_1}}\overline{A}\overline{B}),\end{aligned}$$

which is equivalent to

$$\begin{aligned}\nu(\psi_S) &\equiv A_{s_1}(A \vee B) \wedge (A_{s_2}A \vee \overline{A_{s_2}}\overline{A}) \\ &\equiv A_{s_1}A_{s_2}A(B \vee \overline{B}) \vee A_{s_1}\overline{A_{s_2}}\overline{A}B.\end{aligned}$$

Therewith, we obtain

$$\mathsf{AFOMC}^{\rho}_{\Sigma'}(\nu(\psi_S)) = 2 \cdot s_1 s_2 + s_1 = \mathsf{TMC}_{\Sigma}(\psi_S),$$

where $\rho(A_{s_1}) = s_1$, $\rho(A_{s_2}) = s_2$, and $\rho(l) = 1$ for all remaining ground literals l.

We state that the typed model count of a structured sentence $\phi_S \in \mathcal{SL}^S(\Sigma)$ equals the algebraic model count of its translation $\nu(\phi_S) \in \mathcal{RL}^S(\Sigma)$ in form of a formal proposition.

Proposition 6.3.3: Typed Versus Algebraic Model Counting
Let $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ be a finite signature, let $\phi_S \in \mathcal{SL}^S(\Sigma)$ be a structured sentence, without loss of generality suppose that ϕ_S is quantifier-free, and let

$$\rho\colon \mathsf{grLit}(\Sigma) \cup \{A_s \mid s \in \mathcal{S}_{\mathfrak{G}}\} \cup \{\overline{A_s} \mid s \in \mathcal{S}_{\mathfrak{G}}\} \to \mathcal{S}$$

be a mapping with $\rho(l) = 1$ for all $l \in \mathsf{grLit}(\Sigma)$ and $\rho(A_s) = s$ as well as $\rho(\overline{A_s}) = 1$ for all $s \in \mathcal{S}_{\mathfrak{G}}$. Then,

$$\mathsf{TMC}_{\Sigma}(\phi_S) = \mathsf{AFOMC}^{\rho}_{\Sigma'}(\nu(\phi_S)),$$

where $\Sigma' = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma \cup \{A_s/0 \mid s \in \mathcal{S}_{\mathfrak{G}}\})$.

Proof. As discussed above, it is sufficient to show $\mathsf{TMC}_{\Sigma}(s\circ\phi) = \mathsf{AFOMC}^{\rho}_{\Sigma'}(\nu(s\circ\phi))$, where Σ' is Σ extended by the predicate $A_s/0$ and ρ is given by $\rho(A_s) = s$ and $\rho(l) = 1$ for all remaining ground literals $l \in \mathsf{grLit}(\Sigma')$.

First, note that $\mathsf{TMC}_{\Sigma}(\psi) = \mathsf{AFOMC}^{\rho}_{\Sigma}(\nu(\psi))$ holds for all $\psi \in \mathcal{RL}^S(\Sigma)$ because $\psi = \nu(\psi)$ in this case. Therewith, it follows that

$$\begin{aligned}\mathsf{TMC}_{\Sigma}(s \circ \phi) &= s \otimes \mathsf{TMC}_{\Sigma}(\phi) = s \otimes \mathsf{AFOMC}^{\rho}_{\Sigma}(\nu(\phi)) \\ &= s \otimes \mathsf{AFOMC}^{\rho}_{\Sigma}(\phi) = \mathsf{AFOMC}^{\rho}_{\Sigma'}(A_s \wedge \phi) \\ &= \mathsf{AFOMC}^{\rho}_{\Sigma'}(\phi \wedge (A_s\phi \vee \overline{A_s}\overline{\phi})) = \mathsf{AFOMC}^{\rho}_{\Sigma'}(\nu(s \circ \phi)). \qquad \square\end{aligned}$$

From the connection between typed model counting and algebraic model counting we gain the possibility to exploit counting techniques and further results from algebraic model counting (cf. e.g., [Kimmig et al., 2017]) for typed model counting. On the downside, one could claim that typed model counting is just a variant of algebraic model counting without any benefits but the disadvantage that one has to consider structured sentences. However, there are some advantages of considering structured sentences which is why we prefer typed model counting to algebraic model counting in our concrete setting.

ω	$I_S^\omega(\phi_S)$	ω	$I_S^\omega(\phi_S)$
AB	s^2	$\overline{A}B$	s
$A\overline{B}$	s	$\overline{A}\,\overline{B}$	0

Table 6.5: Structured interpretations of the structured sentence ϕ_S from Example 6.3.4.

ω	$I_\rho^\omega(\nu(\phi_S))$	ω	$I_\rho^\omega(\nu(\phi_S))$
$ABA_sA'_s$	s^2	$\overline{A}BA_sA'_s$	0
$ABA_s\overline{A'_s}$	0	$\overline{A}BA_s\overline{A'_s}$	0
$AB\overline{A_s}A'_s$	0	$\overline{A}B\overline{A_s}A'_s$	s
$AB\overline{A_s}\,\overline{A'_s}$	0	$\overline{A}B\overline{A_s}\,\overline{A'_s}$	0
$A\overline{B}A_sA'_s$	0	$\overline{A}\,\overline{B}A_sA'_s$	0
$A\overline{B}A_s\overline{A'_s}$	s	$\overline{A}\,\overline{B}A_s\overline{A'_s}$	0
$A\overline{B}\,\overline{A_s}A'_s$	0	$\overline{A}\,\overline{B}\,\overline{A_s}A'_s$	0
$A\overline{B}\,\overline{A_s}\,\overline{A'_s}$	0	$\overline{A}\,\overline{B}\,\overline{A_s}\,\overline{A'_s}$	0

Table 6.6: Algebraic interpretations of the sentence ϕ_S from Example 6.3.4 with respect to the assignment ρ as specified in the same example.

▶ Advantages of Typed Model Counting

The main reason why we prefer typed model counting to algebraic model counting here is that typed model counting allows us to multiply count a structure element s if it occurs in a structured sentence ϕ_S several times. Instead, when translating ϕ_S to $\nu(\phi_S)$, then for every occurrence of s a fresh ground atom A_s has to be introduced which results in possibly many duplicates of A_s. This is illustrated best by an example.

Example 6.3.4

We consider the structured sentence

$$\phi_S = (s \circ A) \vee (s \circ B)$$

over the signature $\Sigma = (\emptyset, \{A/0, B/0\})$ with the structure element s. The structured interpretations of ϕ_S are shown in Table 6.5. In order to preserve the multiple counting of the structure element s in the possible world which satisfies both A and B when translating ϕ_S to $\nu(\phi_S)$, it is not sufficient to introduce one fresh ground atom for both occurrences of s only. Instead, one has to introduce one fresh ground atom for each occurrence of s separately. That is, the proper translation of ϕ_S is

$$\nu(\phi_S) = (A \vee B) \wedge (A_s A \vee \overline{A_s}\overline{A}) \wedge (A'_s B \vee \overline{A'_s}\overline{B}),$$

where both A_s and A'_s are mapped by the assignment ρ to s and, for all remaining ground literals l, $\rho(l) = 1$ holds. This means that, on the one hand, the sentence $\nu(\phi_S)$ becomes unnecessarily long and, on the other hand, the number of interpretations of $\nu(\phi_S)$ growths by a factor 2 with each occurrence of s in ϕ_S. Please see Table 6.6 for the algebraic interpretations of $\nu(\phi_S)$.

To be fair, our translation of structured sentences to algebraically interpreted sentences can be improved so that, in some cases, the multiple fresh ground atoms which are introduced by the mapping ν can be aggregated to a ground instances of a common predicate with higher arity. This maintains compact representations as the next example illustrates.

Example 6.3.5

We consider the signature $\Sigma = (\{c_1, \ldots, c_m\}, \{A/1\})$ and the structured sentence

$$\phi_S = \bigvee_{c_i \in \mathsf{Const}_\Sigma} s \circ A(c_i)$$

with structure element s. Instead of introducing a fresh ground atom A_s for every occurrence of the structure element s in ϕ_S, which would result in

$$\nu(\phi_S) = \left(\bigvee_{c_i \in \mathsf{Const}_\Sigma} A(c_i) \right) \wedge (A_s A(c_1) \vee \overline{A_s}\,\overline{A(c_1)}) \wedge (A'_s A(c_2) \vee \overline{A'_s}\,\overline{A(c_2)}) \wedge \ldots,$$

we can introduce one fresh unary predicate $A_s/1$ and translate ϕ_S into the sentence

$$\nu'(\phi_S) = \left(\bigvee_{c \in \mathsf{Const}_\Sigma} A(c) \right) \wedge \bigwedge_{c \in \mathsf{Const}_\Sigma} (A_s(c)A(c) \vee \overline{A_s(c)}\,\overline{A(c)}).$$

This maintains the interchangeability of the constants in Const_Σ. In this sense, ν' can be understood as a "lifted" version of the translation ν.

A more technical reason why we consider typed model counting instead of algebraic model counting here is that algebraic model counting needs with ρ an additional mapping from the ground literals to the algebraic semiring which tells us how to interpret the ground literals. When the goal is to count models efficiently, it is necessary to store and apply ρ efficiently, too. In typed model counting we can concentrate on the analysis of structured sentences instead because the whole information about the typed model counts is encoded in the structured sentence. This is an important point not least because the number of structure elements is relatively small when applying typed model counting to conditional reasoning (cf. Section 6.4), especially compared to the large number of ground literals. The ground literals in structured languages can be handled the same as in classical logics, while in the translation to algebraic model counting there is, in principle, no difference between the initial ground literals and the fresh ground literals that are introduced during the translation process.

A last reason why we consider typed model counting is that, in the context of conditional reasoning, structured sentences in general and structure elements in particular have a clear and accessible meaning (cf. Section 6.4). Structure elements correspond to conditional structures and the experienced knowledge engineer can read from structured sentences which correspond to knowledge bases what part of the structured sentence refers to what conditional in the knowledge base.

6.4 Conditional Knowledge Compilation

In this section, we apply typed model counting to conditional reasoning, in particular to maximum entropy reasoning based on probabilistic conditional knowledge bases. First, we demonstrate how to compute conditional structures with typed model counting, and then we compute the input of the condensed iterative scaling algorithm (cf. Figure 5.2) that can be used to approximate maximum entropy models.

▶ Typed Model Counting and Conditional Structures

Given a finite set of qualitative conditionals $\Delta \subseteq \mathcal{RCL}(\Sigma)$, our goal is to compute the conditional structures of (sets of) possible worlds $\omega \in \Omega(\Sigma)$ with respect to Δ by exploiting the framework of typed model counting (Section 6.2).[2] For this, we have to compile Δ into an adequate structured sentence first.

Let $\mathcal{SL}(\Sigma)$ be the structured language (cf. Definition 6.2.2) defined over the semiring $\mathcal{S}(\Delta)$ which is described in (3.4). Recall that the generators of $\mathcal{S}(\Delta)$ are the algebraic elements a_i^+ and a_i^- for $i = 1, \ldots, n$ with $n = |\Delta|$ which are the constituents of the conditional structures with respect to Δ. The occurrence of the generator a_i^+ in the conditional structure $\sigma_{\Sigma,\Delta}(\omega)$ of a possible world $\omega \in \Omega(\Sigma)$ indicates that an instance of the i-th conditional in Δ is verified in ω, while a_i^-

---

[2] Recall Definition 3.2.1 and Definition 3.2.3 for the definitions of conditional structures of (sets of) possible worlds.

stands for the falsification. With this in mind, we translate Δ as follows.

Definition 6.4.1: Compilation of Conditionals Into Structured Sentences *(cf. [Wilhelm et al., 2019b])*

Let Σ be a finite signature, let $\Delta = \{\delta_1, \ldots, \delta_n\}$ with $\delta_i = (\psi_i|\phi_i)_{\mathsf{CS}_i}$ for $i = 1, \ldots, n$ be a finite set of conditionals over Σ, and let $\mathcal{SL}(\Sigma)$ be the structured language defined over the semiring $\mathcal{S}(\Delta)$. Then, we define the structured sentence $\Phi_\Delta \in \mathcal{SL}^S(\Sigma)$ by

$$\Phi_\Delta = \bigwedge_{i=1}^{n} \Big(\forall_{\mathsf{CS}_i} \mathsf{FreeVar}(\delta_i).\ \big(\mathsf{a}_i^+ \circ \phi_i\psi_i \vee \mathsf{a}_i^- \circ \phi_i\overline{\psi_i} \vee \overline{\phi_i}\big)\Big).$$

The outer conjunction of the formula Φ_Δ mentions for every conditional $\delta_i \in \Delta$ a conjunct which again mentions, by means of the universal quantification, for every ground instance of δ_i a separate disjunction. This disjunction is split into three cases: The verification of the ground instance indicated by a_i^+, the falsification indicated by a_i^-, and the non-applicability. Depending on the structured interpretation $I_{\mathcal{S}}^{\omega}$ in which the structured sentence Φ_Δ is evaluated, exactly one of these three cases is true and the respective structure element (a_i^+, a_i^-, or 1) is collected. As this happens for every ground instance of every conditional in Δ, the structured interpretation of Φ_Δ with respect to $I_{\mathcal{S}}^{\omega}$ returns the conditional structure $\sigma_{\Sigma,\Delta}(\omega)$. This also means that if we perform typed model counting on Φ_Δ, then we obtain $\sigma_{\Sigma,\Delta}(\Omega(\Sigma))$.

Proposition 6.4.2: Typed Model Counting and Conditional Structures

Let Σ be a finite signature, let $\Delta = \{\delta_1, \ldots, \delta_n\}$ with $\delta_i = (\psi_i|\phi_i)_{\mathsf{CS}_i}$ for $i = 1, \ldots, n$ be a finite set of conditionals over Σ, let $\mathcal{SL}(\Sigma)$ be the structured language defined over the semiring $\mathcal{S}(\Delta)$, and let $\Phi_\Delta \in \mathcal{SL}^S(\Sigma)$ be the compilation of Δ into a structured sentence according to Definition 6.4.1. Then,

- $I_{\mathcal{S}}^{\omega}(\Phi_\Delta) = \sigma_{\Sigma,\Delta}(\omega)$ for $\omega \in \Omega(\Sigma)$,
- $\mathsf{TMC}_\Sigma(\Phi_\Delta) = \sigma_{\Sigma,\Delta}(\Omega(\Sigma))$.

Proof. Let $\omega \in \Omega(\Delta)$. Then,

$$\begin{aligned} I_{\mathcal{S}}^{\omega}(\Phi_\Delta) &= I_{\mathcal{S}}^{\omega}(\bigwedge_{i=1}^{n} \Big(\forall_{\mathsf{CS}_i} \mathsf{FreeVar}(\delta_i).\ \big(\mathsf{a}_i^+ \circ \phi_i\psi_i \vee \mathsf{a}_i^- \circ \phi_i\overline{\psi_i} \vee \overline{\phi_i}\big)\Big)) \\ &= \prod_{i=1}^{n} I_{\mathcal{S}}^{\omega}(\forall_{\mathsf{CS}_i} \mathsf{FreeVar}(\delta_i).\ \big(\mathsf{a}_i^+ \circ \phi_i\psi_i \vee \mathsf{a}_i^- \circ \phi_i\overline{\psi_i} \vee \overline{\phi_i}\big)) \end{aligned}$$

$$
\begin{aligned}
&= \prod_{i=1}^{n} I_S^{\omega}\Big(\bigwedge_{(\psi'|\phi')\in\mathsf{Inst}_\Sigma(\delta_i)} (\mathsf{a}_i^+ \circ \phi'_i\psi'_i \vee \mathsf{a}_i^- \circ \phi'_i\overline{\psi'_i} \vee \overline{\phi'_i})\Big) \\
&= \prod_{i=1}^{n} \prod_{(\psi'|\phi')\in\mathsf{Inst}_\Sigma(\delta_i)} I_S^{\omega}(\mathsf{a}_i^+ \circ \phi'_i\psi'_i \vee \mathsf{a}_i^- \circ \phi'_i\overline{\psi'_i} \vee \overline{\phi'_i}) \\
&= \prod_{i=1}^{n} \prod_{(\psi'|\phi')\in\mathsf{Inst}_\Sigma(\delta_i)} \begin{cases} \mathsf{a}_i^+ & \text{if} \quad \omega \models \phi'_i\psi'_i \\ \mathsf{a}_i^- & \text{if} \quad \omega \models \phi'_i\overline{\psi'_i} \\ 1 & \text{if} \quad \omega \models \overline{\phi'_i} \end{cases} \\
&= \prod_{i=1}^{n} \prod_{(\psi'|\phi')\in\mathsf{Inst}_\Sigma(\delta_i)} \sigma_{\Sigma,(\psi'|\phi')}(\omega) \\
&= \prod_{i=1}^{n} \sigma_{\Sigma,\delta_i}(\omega) = \sigma_{\Sigma,\Delta}(\omega),
\end{aligned}
$$

so that the first statement of this proposition is proven. For the proof of the second statement we note that by construction of Φ_Δ every structured interpretation I_S^ω is a typed model of Φ_Δ. This is important for step $(*)$ in the following chain of equations and particularly holds because for every ground instance of each conditional besides the verification and the falsification of this ground instance also its non-applicability is considered in Φ_Δ. Thus,

$$
\begin{aligned}
\mathsf{TMC}_\Sigma(\Phi_\Delta) &= \sum_{I_S^\omega \in \mathsf{Mod}_\Sigma^S(\Phi_\Delta)} I_S^\omega(\Phi_\Delta) = \sum_{I_S^\omega \in \mathsf{Mod}_\Sigma^S(\Phi_\Delta)} \sigma_{\Sigma,\Delta}(\omega) \\
&\overset{(*)}{=} \sum_{\omega\in\Omega(\Sigma)} \sigma_{\Sigma,\Delta}(\omega) = \sigma_{\Sigma,\Delta}(\Omega(\Sigma)).
\end{aligned}
$$

□

We give an example which demonstrates the potential of computing conditional structures with typed model counting well. Actually, we are able to compute the conditional structure in this example in a lifted manner, i.e., with grounding only if it is necessary and with the domain size as a parameter.

Example 6.4.3

We consider the signature $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ with $a \in \mathsf{Const}_\Sigma$, $k = |\mathsf{Const}_\Sigma|$ where $k \geq 1$, and $\mathsf{Pred}_\Sigma = \{F/1, M/1\}$. Thereby, a stands for “Alice,” F stands for “famous,” and M stands for “millionaire.” Further, let $\Delta = \{\delta_1, \delta_2\}$ with

$$
\delta_1 = (F(X)|M(X)), \qquad \delta_2 = (M(a)|\top),
$$

state that “millionaires are usually famous” (δ_1) and “Alice is plausibly a mil-

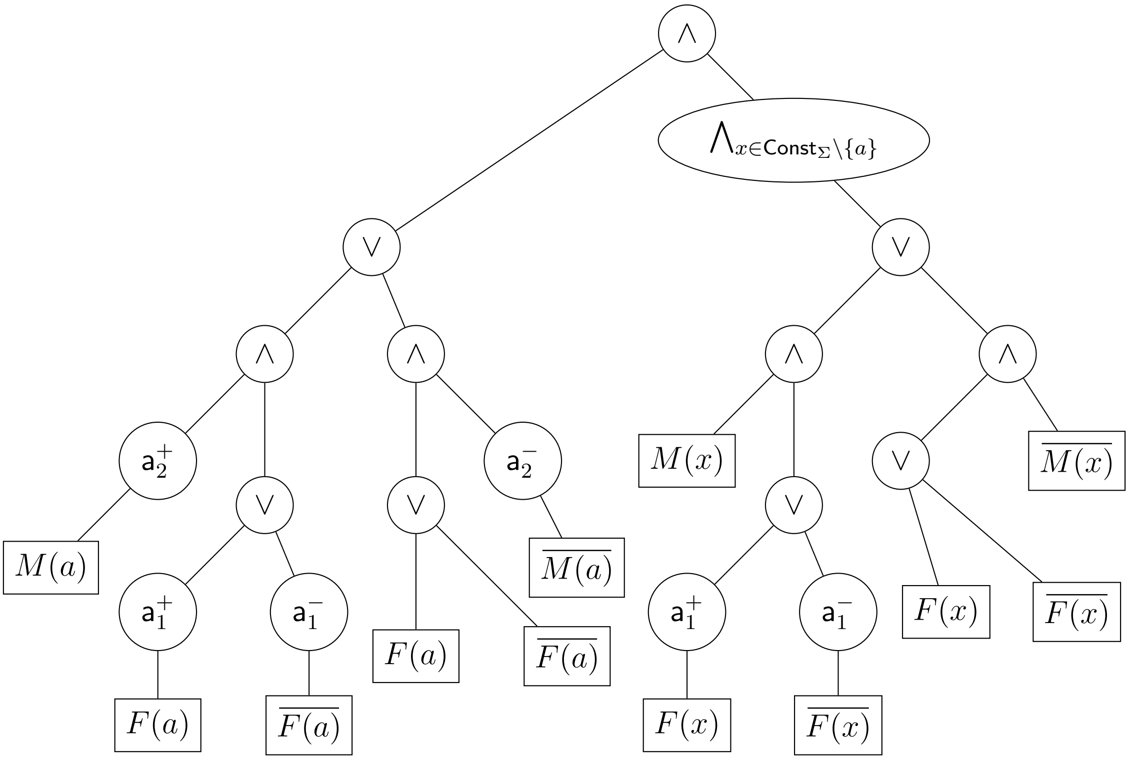

Figure 6.9: sd-DNNF$^{\mathcal{S}}$-circuit equivalent to Φ_Δ from Example 6.4.3.

lionaire" (δ_2). Then, the structured sentence Φ_Δ is (cf. Definition 6.4.1)

$$\begin{aligned}\Phi_\Delta = \forall X.(&\mathsf{a}_1^+ \circ M(X)F(X) \vee \mathsf{a}_1^- \circ M(X)\overline{F(X)} \vee \overline{M(X)})\\ &\wedge (\mathsf{a}_2^+ \circ M(a) \vee \mathsf{a}_2^- \circ \overline{M(a)} \vee \bot).\end{aligned}$$

This structured sentence can be compiled into sd-DNNF$^{\mathcal{S}}$-normal form as shown in the sd-DNNF$^{\mathcal{S}}$-circuit in Figure 6.9 and translated to the structured counting graph in Figure 6.10. Basically, hereby we applied the compilation technique *shattered compilation* and split of expressions with respect to the constant a because it does not behave isomorphic to the other constants. From the counting graph we obtain the typed model count $\mathsf{TMC}_\Sigma(\Phi_\Delta)$ which is

$$\mathsf{TMC}_\Sigma(\Phi_\Delta) = (\mathsf{a}_2^+ \cdot (\mathsf{a}_1^+ + \mathsf{a}_1^-) + 2 \cdot \mathsf{a}_2^-) \cdot (\mathsf{a}_1^+ + \mathsf{a}_1^- + 2)^{k-1}.$$

According to Proposition 6.4.2, $\mathsf{TMC}_\Sigma(\Phi_\Delta)$ equals $\sigma_{\Sigma,\Delta}(\Delta(\Sigma))$. In this compact representation of $\sigma_{\Sigma,\Delta}(\Delta(\Sigma))$ the influence of the constant a (Alice)

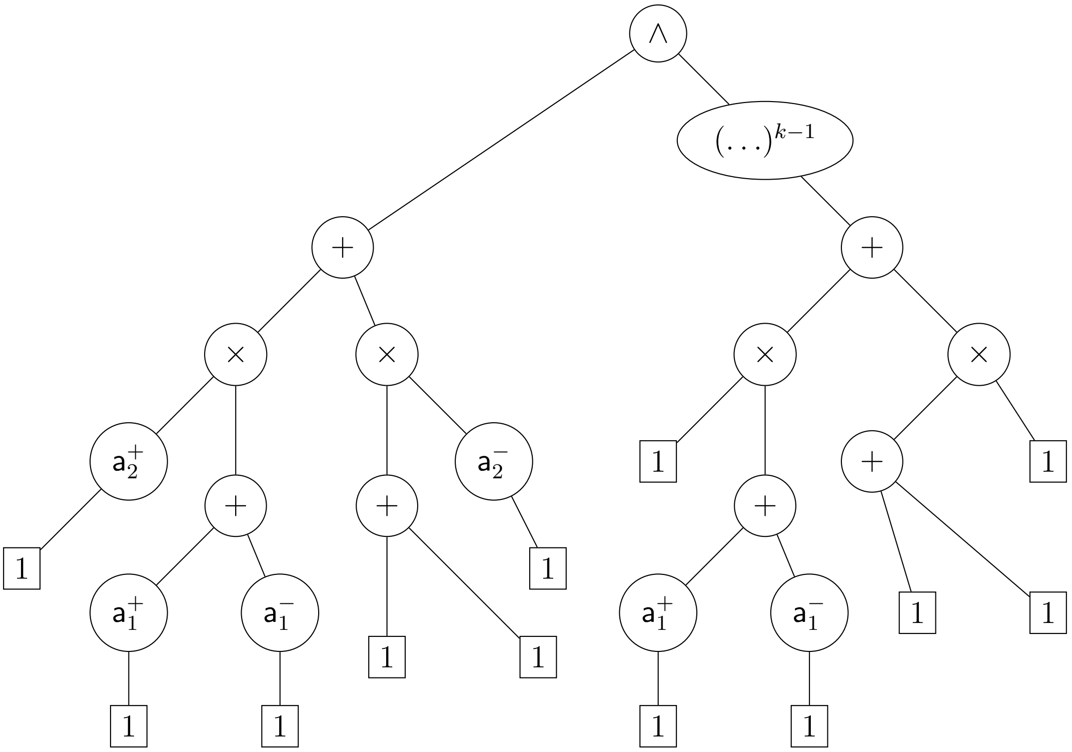

Figure 6.10: Structured counting graph regarding the sd-DNNF$^{\mathcal{S}}$-circuit in Figure 6.9.

on $\sigma_{\Sigma,\Delta}(\Delta(\Sigma))$ is clearly reflected. The factor

$$(\mathsf{a}_2^+ \cdot (\mathsf{a}_1^+ + \mathsf{a}_1^-) + 2 \cdot \mathsf{a}_2^-)$$

in $\sigma_{\Sigma,\Delta}(\Delta(\Sigma))$ refers to the evaluation of the ground instances of the conditionals in Δ with respect to the constant a. Also, the interchangeability of the remaining constants from Const_Σ can be realized, as the factor

$$(\mathsf{a}_1^+ + \mathsf{a}_1^- + 2)^{k-1}$$

refers to the constants other than a.

Next, we transfer our approach to the computation of the input of the condensed iterative scaling algorithm.

▶ Typed Model Counting for Maximum Entropy Reasoning

Conditional structures influence maximum entropy models essentially (cf. Proposition 4.3.4) but they cannot be used to compute maximum entropy models directly. The condensed iterative scaling algorithm CIS presented in Section 5.2, for instance, expects the polynomials (cf. (5.1) and (5.2))

$$\tau_{\Sigma,\mathcal{R}}(x_1,\dots,x_n,y_1,\dots,y_n) = \sum_{\omega\in\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} \prod_{j=1}^{n} x_j^{\mathsf{ver}_{\Sigma,r_j}(\omega)} \cdot y_j^{\mathsf{app}_{\Sigma,r_j}(\omega)}$$

and

$$\tau^{r_i}_{\Sigma,\mathcal{R}}(x_1,\dots,x_n,y_1,\dots,y_n,z) = \sum_{\omega\in\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} (\mathsf{ver}_{\Sigma,r_i}(\omega) + z\cdot \mathsf{app}_{\Sigma,r_i}(\omega)) \cdot \prod_{j=1}^{n} x_j^{\mathsf{ver}_{\Sigma,r_j}(\omega)} \cdot y_j^{\mathsf{app}_{\Sigma,r_j}(\omega)}$$

for $r_i \in \mathcal{B}_\mathcal{R}$, where $\mathcal{R} = (\mathcal{F}_\mathcal{R}, \mathcal{B}_\mathcal{R})$ is a Σ-p-consistent knowledge base. Here, we show how these polynomials can be computed using typed model counting as well. Therewith, we can approximate the maximum entropy Σ-model of $\mathcal{R}$ as follows:

1. Given a Σ-p-consistent knowledge base $\mathcal{R}$, compute $\tau_{\Sigma,\mathcal{R}}$ and $\tau^{r_i}_{\Sigma,\mathcal{R}}$ for $r_i \in \mathcal{B}_\mathcal{R}$ via typed model counting.

2. Initialize the condensed iterative scaling algorithm CIS (5.2) with the results of typed model counting and approximate the ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$ by the output $\vec{\alpha}^*_{\Sigma,\mathcal{R}}$ of CIS.

3. Compute an approximation of the maximum entropy Σ-model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$ of $\mathcal{R}$ via (cf. (5.7))

$$\mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}(\omega) = \begin{cases} \alpha_0^* \cdot \prod_{i=1}^{n} (\alpha_i^*)^{f_{\Sigma,i}(\omega)} & \text{if} \quad \omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma) \\ 0 & \text{otherwise} \end{cases}.$$

After these steps and based on $\mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}$ inductive inferences can be drawn. Alternatively, the vector $\vec{\alpha}^*_{\Sigma,\mathcal{R}}$ can be used to draw inferences directly without computing $\mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}$. We demonstrate by means of an example how this is possible with typed model counting as well.

In order to compute the polynomials $\tau_{\Sigma,\mathcal{R}}$ and $\tau^{r_i}_{\Sigma,\mathcal{R}}$ for $r_i \in \mathcal{B}_\mathcal{R}$ with typed model counting, we instantiate the structured language $\mathcal{SL}(\Sigma)$ with the Abelian semiring $\mathcal{S}_\tau(\mathcal{R}) = (\mathcal{T}_\mathcal{R}, +, \cdot, 0, 1)$ defined over the generating set

$$\mathcal{T}_\mathcal{R} = \{x_1,\dots,x_n,y_1,\dots,y_n\}.$$

The set $\mathcal{T}_{\mathcal{R}}$ mentions, apart from z, the variables which occur in the polynomials $\tau_{\Sigma,\mathcal{R}}$ and $\tau^{r_i}_{\Sigma,\mathcal{R}}$. Therewith, the interpretation of a structured sentence in $\mathcal{SL}^S(\Sigma)$ is a polynomial in the polynomial ring $\mathbb{Z}[x_1,\dots,x_n,y_1,\dots,y_n]$, and we can use the following compilation of knowledge bases into structured sentences.

Definition 6.4.4: Knowledge Base Compilation

(cf. [Wilhelm et al., 2019b])

Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a knowledge base with $\mathcal{B}_{\mathcal{R}} = \{r_1,\dots,r_n\}$ and $r_i = (\psi_i|\phi_i)_{\mathsf{CS}_i}[p_i]$ for $i = 1,\dots,n$, and let $\mathcal{SL}(\Sigma)$ be the structured language defined over the semiring $\mathcal{S}_\tau(\mathcal{R})$. Then, we define the structured sentence $\Psi_{\mathcal{R}} \in \mathcal{SL}^S(\Sigma)$ by

$$\Psi_{\mathcal{R}} = \left(\bigwedge_{F\in\mathcal{F}_{\mathcal{R}}} F\right) \wedge \bigwedge_{i=1}^{n} \Big(\forall_{\mathsf{CS}_i}\mathsf{FreeVar}(r_i).\, \big(y_i \circ \phi_i \wedge (x_i \circ \psi_i \vee \overline{\psi_i}) \vee \overline{\phi_i}\big)\Big).$$

The structured sentence $\psi_{\mathcal{R}}$ is similar to the structured sentence $\phi_{\mathcal{R}}$ from Definition 6.4.1. For every (probabilistic) conditional $r_i \in \mathcal{B}_{\mathcal{R}}$, there is a conjunct in $\psi_{\mathcal{R}}$ which mentions, by means of the universal quantification, for every ground instance of r_i a disjunction. This disjunction slightly differs from the respective disjunction in $\phi_{\mathcal{R}}$, though. Besides that we replaced the structure elements a_i^+ and a_i^- by x_i and y_i, we have rearranged the terms in the disjunction because the structure elements x_i and y_i shall refer to the verification and *applicability* of the ground instance of r_i while the structure elements in $\phi_{\mathcal{R}}$ refer to the verification and *falsification*. In addition, we added the term $\bigwedge_{F\in\mathcal{F}_{\mathcal{R}}} F$ because only those structured interpretations $I^\omega_{\mathcal{S}}$ shall be typed models of $\psi_{\mathcal{R}}$ which refer to possible worlds ω in $\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$. Recall that the "impossible worlds" $\omega \in \Omega(\Sigma) \setminus \Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)$ do not contribute to the maximum entropy model $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}$. We have the following proposition.

Proposition 6.4.5: Typed Model Counting and Maximum Entropy

Let Σ be a finite signature, let $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ be a knowledge base with $\mathcal{B}_{\mathcal{R}} = \{r_1,\dots,r_n\}$ and $r_i = (\psi_i|\phi_i)_{\mathsf{CS}_i}[p_i]$ for $i = 1,\dots,n$, let $\mathcal{SL}(\Sigma)$ be the structured language defined over the semiring $\mathcal{S}_\tau(\mathcal{R})$, and let $\Psi_{\mathcal{R}} \in \mathcal{SL}^S(\Sigma)$ be the compilation of $\mathcal{R}$ into a structured sentence according to Definition 6.4.4. Then,

- $\mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}}) = \tau_{\Sigma,\mathcal{R}}$,
- $\sum_{(\psi_i'|\phi_i')[p_i]\in\mathsf{Inst}_\Sigma(r_i)}(\mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}} \wedge \phi_i' \wedge \psi_i') + z \cdot \mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}} \wedge \phi_i')) = \tau^{r_i}_{\Sigma,\mathcal{R}}$ for $r_i \in \mathcal{B}_{\mathcal{R}}$.

Proof. One has

$$
\begin{aligned}
&\mathsf{TMC}_\Sigma(\Psi_\mathcal{R})\\
&= \sum_{\omega\in\Omega(\Sigma)} I_S^\omega(\Psi_\mathcal{R})\\
&= \sum_{\omega\in\Omega(\Sigma)} I_S^\omega\big(\bigwedge_{F\in\mathcal{F}_\mathcal{R}} F\big)\cdot I_S^\omega\Big(\bigwedge_{i=1}^{n}\Big(\forall_{\mathsf{CS}_i}\mathsf{FreeVar}(r_i).\,\big(y_i\circ\phi_i\wedge(x_i\circ\psi_i\vee\overline{\psi_i})\vee\overline{\phi_i}\big)\Big)\Big)\\
&= \sum_{\omega\in\Omega^0_{\mathcal{F}_\mathcal{R}}(\Sigma)} I_S^\omega\big(\bigwedge_{F\in\mathcal{F}_\mathcal{R}} F\big)\cdot I_S^\omega\Big(\bigwedge_{i=1}^{n}\Big(\forall_{\mathsf{CS}_i}\mathsf{FreeVar}(r_i).\,\big(y_i\circ\phi_i\wedge(x_i\circ\psi_i\vee\overline{\psi_i})\vee\overline{\phi_i}\big)\Big)\Big)\\
&\quad+\sum_{\omega\in\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} I_S^\omega\big(\bigwedge_{F\in\mathcal{F}_\mathcal{R}} F\big)\cdot I_S^\omega\Big(\bigwedge_{i=1}^{n}\Big(\forall_{\mathsf{CS}_i}\mathsf{FreeVar}(r_i).\,\big(y_i\circ\phi_i\wedge(x_i\circ\psi_i\vee\overline{\psi_i})\vee\overline{\phi_i}\big)\Big)\Big)\\
&= \sum_{\omega\in\Omega^0_{\mathcal{F}_\mathcal{R}}(\Sigma)} 0\cdot I_S^\omega\Big(\bigwedge_{i=1}^{n}\Big(\forall_{\mathsf{CS}_i}\mathsf{FreeVar}(r_i).\,\big(y_i\circ\phi_i\wedge(x_i\circ\psi_i\vee\overline{\psi_i})\vee\overline{\phi_i}\big)\Big)\Big)\\
&\quad+\sum_{\omega\in\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} 1\cdot I_S^\omega\Big(\bigwedge_{i=1}^{n}\Big(\forall_{\mathsf{CS}_i}\mathsf{FreeVar}(r_i).\,\big(y_i\circ\phi_i\wedge(x_i\circ\psi_i\vee\overline{\psi_i})\vee\overline{\phi_i}\big)\Big)\Big)\\
&= \sum_{\omega\in\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} I_S^\omega\Big(\bigwedge_{i=1}^{n}\Big(\forall_{\mathsf{CS}_i}\mathsf{FreeVar}(r_i).\,\big(y_i\circ\phi_i\wedge(x_i\circ\psi_i\vee\overline{\psi_i})\vee\overline{\phi_i}\big)\Big)\Big)\\
&= \sum_{\omega\in\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} \prod_{i=1}^{n} I_S^\omega\big(\forall_{\mathsf{CS}_i}\mathsf{FreeVar}(r_i).\,\big(y_i\circ\phi_i\wedge(x_i\circ\psi_i\vee\overline{\psi_i})\vee\overline{\phi_i}\big)\big)\\
&= \sum_{\omega\in\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} \prod_{i=1}^{n} \prod_{(\psi'_i|\phi'_i)[p_i]\in\mathsf{Inst}_\Sigma(r_i)} I_S^\omega(y_i\circ\phi'_i\wedge(x_i\circ\psi'_i\vee\overline{\psi'_i})\vee\overline{\phi'_i})\\
&= \sum_{\omega\in\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} \prod_{i=1}^{n} \prod_{(\psi'_i|\phi'_i)[p_i]\in\mathsf{Inst}_\Sigma(r_i)} \begin{cases} y_i\cdot x_i & \omega\models\phi'_i\psi'_i\\ y_i & \omega\models\phi'_i\overline{\psi'_i}\\ 1 & \omega\models\overline{\phi'_i}\end{cases}\\
&= \sum_{\omega\in\Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} \prod_{i=1}^{n} x_i^{\mathsf{ver}_{\Sigma,r_i}(\omega)}\cdot y_i^{\mathsf{app}_{\Sigma,r_i}(\omega)}\\
&= \tau_{\Sigma,\mathcal{R}}.
\end{aligned}
$$

Analogously, one has, for $r_i\in\mathcal{B}_\mathcal{R}$,

$$
\begin{aligned}
&\sum_{(\psi'_i|\phi'_i)[p_i]\in\mathsf{Inst}_\Sigma(r_i)} \big(\mathsf{TMC}_\Sigma(\Psi_\mathcal{R}\wedge\phi'_i\wedge\psi'_i)+z\cdot\mathsf{TMC}_\Sigma(\Psi_\mathcal{R}\wedge\phi'_i)\big)\\
={}&\sum_{(\psi'_i|\phi'_i)[p_i]\in\mathsf{Inst}_\Sigma(r_i)} \Big(\sum_{\omega\in\Omega(\Sigma)} I_S^\omega(\Psi_\mathcal{R}\wedge\phi'_i\wedge\psi'_i)+z\cdot\sum_{\omega\in\Omega(\Sigma)} I_S^\omega(\Psi_\mathcal{R}\wedge\phi'_i)\Big)
\end{aligned}
$$

$$\begin{aligned}
&= \sum_{(\psi_i'|\phi_i')[p_i] \in \mathsf{Inst}_\Sigma(r_i)} \Big(\sum_{\omega \in \Omega(\Sigma)\colon \omega \models_\Sigma \phi_i'\psi_i'} I_S^\omega(\Psi_\mathcal{R}) + z \cdot \sum_{\omega \in \Omega(\Sigma)\colon \omega \models_\Sigma \phi_i'} I_S^\omega(\Psi_\mathcal{R}) \Big) \\
&= \sum_{(\psi_i'|\phi_i')[p_i] \in \mathsf{Inst}_\Sigma(r_i)} \; \sum_{\omega \in \Omega(\Sigma)\colon \omega \models_\Sigma \phi_i'\psi_i'} I_S^\omega(\Psi_\mathcal{R}) \\
&\quad + z \cdot \sum_{(\psi_i'|\phi_i')[p_i] \in \mathsf{Inst}_\Sigma(r_i)} \; \sum_{\omega \in \Omega(\Sigma)\colon \omega \models_\Sigma \phi_i'} I_S^\omega(\Psi_\mathcal{R}) \\
&= \sum_{(\psi_i'|\phi_i')[p_i] \in \mathsf{Inst}_\Sigma(r_i)} \; \sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)\colon \omega \models_\Sigma \phi_i'\psi_i'} \prod_{j=1}^{n} x_j^{\mathsf{ver}_{\Sigma,r_j}(\omega)} \cdot y_j^{\mathsf{app}_{\Sigma,r_j}(\omega)} \\
&\quad + z \cdot \sum_{(\psi_i'|\phi_i')[p_i] \in \mathsf{Inst}_\Sigma(r_i)} \; \sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)\colon \omega \models_\Sigma \phi_i'} \prod_{j=1}^{n} x_j^{\mathsf{ver}_{\Sigma,r_j}(\omega)} \cdot y_j^{\mathsf{app}_{\Sigma,r_j}(\omega)} \\
&= \sum_{(\psi_i'|\phi_i')[p_i] \in \mathsf{Inst}_\Sigma(r_i)} \; \sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)\colon \omega \models_\Sigma \phi_i'\psi_i'} \prod_{j=1}^{n} x_j^{\mathsf{ver}_{\Sigma,r_j}(\omega)} \cdot y_j^{\mathsf{app}_{\Sigma,r_j}(\omega)} \\
&\quad + z \cdot \sum_{(\psi_i'|\phi_i')[p_i] \in \mathsf{Inst}_\Sigma(r_i)} \; \sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)\colon \omega \models_\Sigma \phi_i'} \prod_{j=1}^{n} x_j^{\mathsf{ver}_{\Sigma,r_j}(\omega)} \cdot y_j^{\mathsf{app}_{\Sigma,r_j}(\omega)} \\
&= \sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} \; \sum_{(\psi_i'|\phi_i')[p_i] \in \mathsf{Inst}_\Sigma(r_i)} \mathbb{1}_{\omega \models_\Sigma \phi_i'\psi_i'} \cdot \prod_{j=1}^{n} x_j^{\mathsf{ver}_{\Sigma,r_j}(\omega)} \cdot y_j^{\mathsf{app}_{\Sigma,r_j}(\omega)} \\
&\quad + z \cdot \sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} \; \sum_{(\psi_i'|\phi_i')[p_i] \in \mathsf{Inst}_\Sigma(r_i)} \mathbb{1}_{\omega \models_\Sigma \phi_i'\psi_i'} \cdot \prod_{j=1}^{n} x_j^{\mathsf{ver}_{\Sigma,r_j}(\omega)} \cdot y_j^{\mathsf{app}_{\Sigma,r_j}(\omega)} \\
&= \sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} \mathsf{ver}_{\Sigma,r_i} \cdot \prod_{j=1}^{n} x_j^{\mathsf{ver}_{\Sigma,r_j}(\omega)} \cdot y_j^{\mathsf{app}_{\Sigma,r_j}(\omega)} \\
&\quad + z \cdot \sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} \mathsf{app}_{\Sigma,r_i} \cdot \prod_{j=1}^{n} x_j^{\mathsf{ver}_{\Sigma,r_j}(\omega)} \cdot y_j^{\mathsf{app}_{\Sigma,r_j}(\omega)} \\
&= \sum_{\omega \in \Omega_{\mathcal{F}_\mathcal{R}}(\Sigma)} \big(\mathsf{ver}_{\Sigma,r_i} + z \cdot \mathsf{app}_{\Sigma,r_i}\big) \cdot \prod_{j=1}^{n} x_j^{\mathsf{ver}_{\Sigma,r_j}(\omega)} \cdot y_j^{\mathsf{app}_{\Sigma,r_j}(\omega)} \\
&= \mathcal{T}_{\Sigma,\mathcal{R}}^{r_i}.
\end{aligned}$$

Hereby, $\mathbb{1}_{\omega \models_\Sigma \phi}$ is the indicator function defined by

$$\mathbb{1}_{\omega \models_\Sigma \phi} = \begin{cases} 1 & \text{if} \quad \omega \models_\Sigma \phi \\ 0 & \text{if} \quad \omega \models_\Sigma \overline{\phi} \end{cases}$$

for possible worlds $\omega \in \Omega(\Sigma)$ and sentences $\phi \in \mathcal{RL}^S(\Sigma)$. □

We give an example. More complex examples with several conditionals and with quantifiers can be found in Section A.2 in the appendix.

Example 6.4.6
We consider the signature $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ with $a \in \mathsf{Const}_\Sigma$, $k = |\mathsf{Const}_\Sigma|$ where $k \geq 1$, and $\mathsf{Pred}_\Sigma = \{F/1, M/1\}$, as well as the knowledge base $\mathcal{R} = (\{M(a)\}, \{(F(X)|M(X))[p]\})$ with $p \in (0,1)$ (cf. the similarity to Example 6.4.3). Then,

$$\Psi_\mathcal{R} = M(a) \wedge \forall X.(y_1 \circ M(X) \wedge (x_1 \circ F(X) \vee \overline{F(X)}) \vee \overline{M(X)}).$$

A structured sentence which is $\mathcal{S}$-equivalent to $\Psi_\mathcal{R}$ and which is in $\mathsf{sd\text{-}DNNF}^\mathcal{S}$-normal form is

$$\begin{aligned}\Psi'_\mathcal{R} = {} & y_1 \circ M(a)(x_1 \circ F(a) \vee \overline{F(a)}) \\ & \wedge \bigwedge_{c \in \mathsf{Const}_\Sigma \setminus \{a\}} (y_1 \circ M(c) \wedge (x_1 \circ F(c) \vee \overline{F(c)}) \vee \overline{M(c)} \wedge (F(c) \vee \overline{F(c)})).\end{aligned}$$

Therewith, we obtain

$$\tau_{\Sigma,\mathcal{R}} = \mathsf{TMC}_\Sigma(\Psi_\mathcal{R}) = \mathsf{TMC}_\Sigma(\Psi'_\mathcal{R}) = y_1(x_1+1)(y_1(x_1+1)+2)^{k-1}.$$

For $\tau^{r_1}_{\Sigma,\mathcal{R}}$ we proceed as follows. First, we consider the ground instance of r_1 with respect to the only constant which occurs in $\mathcal{R}$ explicitly, namely the ground instance $r_1^a = (F(a)|M(a))[p]$. Then,

$$\Psi_\mathcal{R} \wedge M(a) \equiv_\mathcal{S} \Psi'_\mathcal{R}$$

and

$$\begin{aligned}\Psi_\mathcal{R} \wedge F(a) \wedge M(a) \equiv_\mathcal{S} {} & y_1 x_1 \circ M(a)F(a) \\ & \wedge \bigwedge_{c \in \mathsf{Const}_\Sigma \setminus \{a\}} (y_1 \circ M(c)(x_1 \circ F(c) \vee \overline{F(c)}) \vee \overline{M(c)}(F(c) \vee \overline{F(c)})),\end{aligned}$$

so that

$$\begin{aligned}\mathsf{TMC}_\Sigma(\Psi_\mathcal{R} \wedge M(a)) &= y_1 \cdot (x_1+1) \cdot (y_1(x_1+1)+2)^{k-1}, \\ \mathsf{TMC}_\Sigma(\Psi_\mathcal{R} \wedge M(a) \wedge F(a)) &= y_1 x_1 \cdot (y_1(x_1+1)+2)^{k-1}.\end{aligned}$$

Now, we consider a ground instance of r_1 with respect to any unnamed constant $b \in \mathsf{Const}_\Sigma$ that does not occur in $\mathcal{R}$ explicitly, i.e., $r_1^b = (F(b)|M(b))[p]$, provided that $k > 1$ so that such a constant exists. Then,

$$\begin{aligned}\Psi_{\mathcal{R}} \wedge M(b) \equiv_{\mathcal{S}} & \Big(y_1 \circ M(a)(x_1 \circ F(a) \vee \overline{F(a)})\Big) \wedge \Big(y_1 \circ M(b)(x_1 \circ F(b) \vee \overline{F(b)})\Big) \\ & \wedge \bigwedge_{c \in \mathsf{Const}_\Sigma \setminus \{a,b\}} (y_1 \circ M(c)(x_1 \circ F(c) \vee \overline{F(c)}) \vee \overline{M(c)}(F(c) \vee \overline{F(c)})),\end{aligned}$$

and

$$\begin{aligned}\Psi_{\mathcal{R}} \wedge M(b) \wedge F(b) \equiv_{\mathcal{S}} & \Big(y_1 \circ M(a)(x_1 \circ F(a) \vee \overline{F(a)})\Big) \wedge (y_1 x_1 \circ M(b)F(b)) \\ & \wedge \bigwedge_{c \in \mathsf{Const}_\Sigma \setminus \{a,b\}} (y_1 \circ M(c)(x_1 \circ F(c) \vee \overline{F(c)}) \vee \overline{M(c)}(F(c) \vee \overline{F(c)})),\end{aligned}$$

so that

$$\begin{aligned}\mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}} \wedge M(b)) &= (y_1(x_1+1))^2 \cdot (y_1(x_1+1)+2)^{k-2}, \\ \mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}} \wedge M(b) \wedge F(b)) &= (y_1(x_1+1)) \cdot y_1 x_1 \cdot (y_1(x_1+1)+2)^{k-2}.\end{aligned}$$

When we stick these findings together, we obtain

$$\begin{aligned}\tau^{r_1}_{\Sigma,\mathcal{R}} &= \sum_{(\psi'_1|\phi'_1)[p_1] \in \mathsf{Inst}_\Sigma(r_1)} (\mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}} \wedge \phi'_1 \wedge \psi'_1) + z \cdot \mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}} \wedge \phi'_1)) \\ &= (\mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}} \wedge M(a) \wedge F(a)) + z \cdot \mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}} \wedge M(a))) \\ &\quad + \sum_{b \in \mathsf{Const}_\Sigma \setminus \{a\}} (\mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}} \wedge M(b) \wedge F(b)) + z \cdot \mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}} \wedge M(b))) \\ &= (y_1 \cdot (x_1+1) \cdot (y_1(x_1+1)+2)^{k-1} + z y_1 x_1 \cdot (y_1(x_1+1)+2)^{k-1}) \\ &\quad + (k-1)((y_1(x_1+1))^2 \cdot (y_1(x_1+1)+2)^{k-2} \\ &\quad + (y_1(x_1+1)) \cdot y_1 x_1 \cdot (y_1(x_1+1)+2)^{k-2})\end{aligned}$$

in case of $k > 1$ and

$$\tau^{r_1}_{\Sigma,\mathcal{R}} = y_1 x_1 + z y_1 (x_1 + 1)$$

in case of $k = 1$.

Finally, we demonstrate by means of an example how to draw inferences via typed model counting from an approximation $\mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}$ of the maximum entropy Σ-model of $\mathcal{R}$. For this, we assume that we have computed the approximation $\vec{\alpha}^*_{\Sigma,\mathcal{R}}$ of the ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}}$ already.

Example 6.4.7

We consider the knowledge base $\mathcal{R} = (\{M(a)\}, \{F(X)|M(X)[p]\})$ with $p \in (0,1)$ from Example 6.4.6, and we assume both $k > 1$ and that an approximation $\vec{\alpha}^*_{\Sigma,\mathcal{R}} = (\alpha^*_1)$ of the Σ-ME-vector $\vec{\alpha}_{\Sigma,\mathcal{R}} = (\alpha_1)$ has already been computed. We ask for the probability q with which $(M(b)|F(b))[q]$ with $b \neq a$

follows from $\mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}$. To answer this query, we compute

$$\begin{aligned} q &= \frac{\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)\colon\, \omega\models F(b)M(b)} \mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}(\omega)}{\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)\colon\, \omega\models F(b)} \mathcal{P}_{\vec{\alpha}^*_{\Sigma,\mathcal{R}}}(\omega)} \\ &= \frac{\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)\colon\, \omega\models F(b)M(b)} (\alpha_1^*)^{f_{\Sigma,1}(\omega)}}{\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)\colon\, \omega\models F(b)} (\alpha_1^*)^{f_{\Sigma,1}(\omega)}} \\ &= \frac{\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)\colon\, \omega\models F(b)M(b)} (\alpha_1^*)^{\mathsf{ver}_{\Sigma,1}(\omega)} \cdot ((\alpha_1^*)^{-p})^{\mathsf{app}_{\Sigma,1}(\omega)}}{\sum_{\omega\in\Omega_{\mathcal{F}_{\mathcal{R}}}(\Sigma)\colon\, \omega\models F(b)} (\alpha_1^*)^{\mathsf{ver}_{\Sigma,1}(\omega)} \cdot ((\alpha_1^*)^{-p})^{\mathsf{app}_{\Sigma,1}(\omega)}}. \end{aligned}$$

The numerator of this fraction can be obtained by computing the typed model count $\mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}} \wedge M(b) \wedge F(b))$ which is a polynomial in x_1 and y_1 in which we have to plug in α_1^* for x_1 and $(\alpha_1^*)^{-p}$ for y_1. Analogously, the denominator can be obtained from $\mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}} \wedge F(b))$. We get (cf. Example 6.4.6)

$$\begin{aligned} \mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}} \wedge M(b) \wedge F(b)) &= (y_1(x_1+1)) \cdot y_1 x_1 \cdot (y_1(x_1+1)+2)^{k-2}, \\ \mathsf{TMC}_\Sigma(\Psi_{\mathcal{R}} \wedge F(b)) &= (y_1(x_1+1)) \cdot (y_1 x_1 + 1) \cdot (y_1(x_1+1)+2)^{k-2}, \end{aligned}$$

so that

$$\begin{aligned} q &= \frac{((\alpha_1^*)^{-p} \cdot (\alpha_1^*+1)) \cdot (\alpha_1^*)^{-p} \cdot \alpha_1^* \cdot ((\alpha_1^*)^{-p}(\alpha_1^*+1)+2)^{k-2}}{((\alpha_1^*)^{-p} \cdot (\alpha_1^*+1)) \cdot ((\alpha_1^*)^{-p} \cdot \alpha_1^* + 1) \cdot ((\alpha_1^*)^{-p} \cdot (\alpha_1^*+1)+2)^{k-2}} \\ &= \frac{(\alpha_1^*)^{-p} \cdot \alpha_1^*}{(\alpha_1^*)^{-p} \cdot \alpha_1^* + 1} \\ &= 1 - \frac{1}{(\alpha_1^*)^{1-p}+1}. \end{aligned}$$

Note that the result seems to be independent of the domain size k. However, this is only partly true because the value of α_1^* depends on k.

To sum up, typed model counting constitutes a framework which allows for a systematic analysis of conditional structures and the computation of the input of the condensed iterative scaling algorithm. How compact this input can be represented and whether the domain size k can be abstracted as a parameter depends on the knowledge base and the applied techniques to compile structured sentences into $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal form. In the next section, we eventually analyze cases in a description logical context in which maximum entropy computations can be performed in a domain-lifted manner.

7

Maximum Entropy and Description Logics

In this chapter, we introduce the probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ which brings together the Description Logic $\mathcal{ALC}$ and relational maximum entropy reasoning (Section 7.1). In Section 7.2 we develop highly specialized methods for maximum entropy reasoning specifically adapted to the $\mathcal{DL}$-context. Because $\mathcal{ALC}^{\mathsf{ME}}$ is defined with respect to a fixed finite domain, the more generic approaches on relational maximum entropy reasoning, which we developed in the previous chapters, can be applied to $\mathcal{ALC}^{\mathsf{ME}}$ as well. Eventually, in Section 7.3, we investigate maximum entropy reasoning for countably infinite domains. We propose an approach on integrating statements about infinite domains into $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases which is based on the concept of satisfiability modulo theory. Therewith, it is possible to formulate statements about infinite domains while the probability space remains finite.

7.1 Description Logic $\mathcal{ALC}^{\mathsf{ME}}$

We define the probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ [Wilhelm et al., 2019a; Wilhelm and Kern-Isberner, 2019; Baader et al., 2019] which integrates probabilistic conditionals into the Description Logic $\mathcal{ALC}$ (cf. Section 2.4). For this, we extend $\mathcal{ALC}$-knowledge bases by a so-called CBox serving as a container for probabilistic conditionals. The TBox and the ABox of classical $\mathcal{ALC}$-knowledge bases remain unchanged such that an $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base is a triple $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ with a clear separation between terminological knowledge in the TBox $\mathcal{T}$, assertions in the ABox $\mathcal{A}$, and uncertain beliefs in the CBox $\mathcal{C}$.

When giving $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases a probabilistic interpretation, we rely on the aggregating semantics (cf. Section 3.4) and the principle of maximum entropy (cf. Chapter 4). The principle of maximum entropy is indicated by the superscript in $\mathcal{ALC}^{\mathsf{ME}}$. Note that we consider fixed finite domains and stick to the *unique name assumption* here [Russell and Norvig, 2010] so that the integration of the aggregating semantics and the principle of maximum entropy works as known from our relational setting. In order to exploit this connection even more, we compile $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases into knowledge bases according to Definition 4.1.1 by extending the embedding of $\mathcal{ALC}^{\mathsf{ffd}}$ into the relational setting (cf. Definition 2.5.1). Therewith, all techniques on systematic maximum entropy reasoning for relational knowledge bases, in particular the concept of *typed model counting* from Section 6.2, can be applied to $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases.

▶ Syntax of $\mathcal{ALC}^{\mathsf{ME}}$

We define the syntax of the probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ and begin with the definition of $\mathcal{ALC}^{\mathsf{ME}}$-concepts (cf. also the definition of $\mathcal{ALC}$-concepts in Definition 2.4.1). For this, let $\mathcal{N}_I$, $\mathcal{N}_C$, and $\mathcal{N}_R$ be disjoint sets of individual, concept and role names, respectively.

Definition 7.1.1: $\mathcal{ALC}^{\mathsf{ME}}$-Concept *(cf. [Wilhelm et al., 2019a])*

An $\mathcal{ALC}^{\mathsf{ME}}$-*concept* is either a concept name $A \in \mathcal{N}_C$ or of the form

$$\top, \quad \bot, \quad \neg C, \quad C \sqcup D, \quad C \sqcap D, \quad \exists r.C, \quad \forall r.C,$$

where C and D are $\mathcal{ALC}^{\mathsf{ME}}$-concepts, and $r \in \mathcal{N}_R$ is a role name.

A *general concept inclusion* (GCI) in $\mathcal{ALC}^{\mathsf{ME}}$ is an expression of the form $C \sqsubseteq D$ where C and D are $\mathcal{ALC}^{\mathsf{ME}}$-concepts. General concept inclusions allow one to express terminological knowledge like "each individual which has property C also has property D" just as in the classical Description Logic $\mathcal{ALC}$. A TBox is a finite set of general concept inclusions.

Further, an $\mathcal{ALC}^{\mathsf{ME}}$-*assertion* is either of the form $C(a)$ ("a has property C") or $r(a,b)$ ("a is related to b via r") where C is an $\mathcal{ALC}^{\mathsf{ME}}$-concept, $r \in \mathcal{N}_{\mathcal{R}}$ is a role name, and $a, b \in \mathcal{N}_I$ are individual names. General concept inclusions and assertions are subsumed under the term *axiom*. A finite set of $\mathcal{ALC}^{\mathsf{ME}}$-assertions is called an ABox. Therewith, the definitions of a TBox and of an ABox in $\mathcal{ALC}^{\mathsf{ME}}$ coincide with the respective definitions in $\mathcal{ALC}$. In addition, we introduce $\mathcal{ALC}^{\mathsf{ME}}$-*conditionals* as follows.

Definition 7.1.2: $\mathcal{ALC}^{\mathsf{ME}}$-Conditional *(cf. [Wilhelm et al., 2019a])*

Let C and D be $\mathcal{ALC}^{\mathsf{ME}}$-concepts, and let $p \in [0,1]$ be a probability. Then, an expression of the form $(D|C)[p]$ is an $\mathcal{ALC}^{\mathsf{ME}}$-*conditional*, or *conditional* for short.

Concept	Syntax	Semantics
Top concept	$\top$	$\Delta^{\mathcal{I}}$
Bottom concept	$\bot$	$\emptyset$
Negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
Conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
Universal restriction	$\forall r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}} : (x,y) \in r^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$
Existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$

Table 7.1: Interpretation of compounded concepts in $\mathcal{ALC}^{\mathsf{ME}}$.

Compared to relational probabilistic conditionals in $\mathcal{RPCL}(\Sigma)$ (cf. Definition 3.3.1), the $\mathcal{ALC}^{\mathsf{ME}}$-conditionals of the form $(D|C)[p]$ have a more specific meaning because the premise C and the conclusion D are limited to $\mathcal{ALC}^{\mathsf{ME}}$-concepts and, thus, refer to properties of individuals. From a technical point of view, we will see that $\mathcal{ALC}^{\mathsf{ME}}$-conditionals $(D|C)[p]$ correspond to relational conditionals $(D(X)|C(X))[p]$ where $C(X), D(X) \in \mathcal{RL}(\Sigma)$ are formulas with one free variable X. In particular, there is one ground instance of the conditional per constant. An $\mathcal{ALC}^{\mathsf{ME}}$-conditional $(D|C)[p]$ can be read as: "If an individual a has property C, then it usually has property D with a probability of p."

Definition 7.1.3: $\mathcal{ALC}^{\mathsf{ME}}$-Knowledge Base *(cf. [Wilhelm et al., 2019a])*

A finite set of $\mathcal{ALC}^{\mathsf{ME}}$-conditionals $(D|C)[p]$ with $p \in (0,1)$ is called a CBox. Let $\mathcal{T}$ be a TBox, let $\mathcal{A}$ be an ABox, and let $\mathcal{C}$ be a CBox. Then, the triple $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ is called an $\mathcal{ALC}^{\mathsf{ME}}$-*knowledge base.*

We restrict $\mathcal{ALC}^{\mathsf{ME}}$-conditionals $(D|C)[p]$ in $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ to be non-factual, i.e., we prohibit probabilities $p \in \{0,1\}$. Therewith, we have a clear separation between terminological knowledge in $\mathcal{T}$, assertional knowledge in $\mathcal{A}$, and uncertain beliefs in $\mathcal{C}$. This is without loss of generality as conditional statements of the form $(D|C)[1]$ or $(D|C)[0]$ can be realized by the general concept inclusions $C \sqsubseteq D$ and $C \sqsubseteq \neg D$, respectively.

Further, $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases do not allow for uncertain assertions. This makes formal arguments simpler, especially when investigating the efficiency of computations. However, in principle, this is not a necessary precondition for our further investigations.

▶ Semantics of $\mathcal{ALC}^{\mathsf{ME}}$

The semantics of $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases is given by probability distributions over $\mathcal{DL}$-interpretations. Recall that a $\mathcal{DL}$*-interpretation* in classical Description Logics is a tuple $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of a non-empty set $\Delta^{\mathcal{I}}$ called *domain* and an *interpretation function* $\cdot^{\mathcal{I}}$ that maps every individual name $a \in \mathcal{N}_I$ to a domain element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, every concept name $C \in \mathcal{N}_C$ to a subset $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ of the domain, and every role name $r \in \mathcal{N}_R$ to a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation of compounded concepts is defined as in $\mathcal{ALC}$ and recalled in Table 7.1. Axioms in $\mathcal{ALC}^{\mathsf{ME}}$ are interpreted as in $\mathcal{ALC}$ on the basis of $\mathcal{DL}$-interpretations.

Definition 7.1.4: $\mathcal{ALC}^{\mathsf{ME}}$-Model *(cf. [Wilhelm et al., 2019a])*

A $\mathcal{DL}$-interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a *(classical)* $\mathcal{ALC}^{\mathsf{ME}}$*-model* of

- a general concept inclusion $C \sqsubseteq D$, denoted by $\mathcal{I} \models C \sqsubseteq D$, iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$,
- an assertion $C(a)$, denote by $\mathcal{I} \models C(a)$, iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$,
- an assertion $r(a, b)$, denoted by $\mathcal{I} \models r(a, b)$, iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$.

$\mathcal{ALC}^{\mathsf{ME}}$-conditionals require a probabilistic interpretation, where the probabilities are understood as degrees of belief. In order to ensure that the $\mathcal{DL}$-interpretations have the same scope and the probability space is well-defined, we assume that all $\mathcal{DL}$-interpretations refer to the same *fixed finite domain*. We also stick to the unique name assumption here, like in our relational setting.

The fixed finite domain assumption demands that each $\mathcal{DL}$-interpretation $\mathcal{I}$ is of the form $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ with the same domain Δ. According to the unique name assumption, in all $\mathcal{DL}$-interpretations the individual names are interpreted by the same domain element. That is, for all $\mathcal{DL}$-interpretations $\mathcal{I}$ and $\mathcal{I}'$ and for all individual names $a, b \in \mathcal{N}_I$, we have $a^{\mathcal{I}} = a^{\mathcal{I}'}$ as well as $a^{\mathcal{I}} = b^{\mathcal{I}}$ iff $a = b$. Also, every domain element is represented by an individual name. As a consequence, we may set $\Delta = \mathcal{N}_I$. Eventually, we may assume that the sets of concept names $\mathcal{N}_C$ and the set of role names $\mathcal{N}_R$ are finite, too ($\mathcal{T}$, $\mathcal{A}$, and $\mathcal{C}$ are all finite). Then, the number of possible $\mathcal{DL}$-interpretations over $\mathcal{N}_I$, $\mathcal{N}_C$, and $\mathcal{N}_R$ is also finite, and we denote the set of all these $\mathcal{DL}$-interpretations by $\mathfrak{I}(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R)$. Because we do not vary $\mathcal{N}_C$ and $\mathcal{N}_R$ unless we explicitly mention it, we usually write $\mathfrak{I}_\Delta$ instead of $\mathfrak{I}(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R)$. For $\mathcal{ALC}^{\mathsf{ME}}$-concepts C and D, we say that C and D are equivalent, denoted by $C \equiv D$, iff $C^{\mathcal{I}} = D^{\mathcal{I}}$ holds for all interpretations $\mathcal{I} \in \mathfrak{I}_\Delta$. *Probabilistic models* of $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases are specific probability distributions over $\mathfrak{I}_\Delta$ then.

Definition 7.1.5: Probabilistic $\mathcal{ALC}^{\mathsf{ME}}$-Model

(cf. [Wilhelm et al., 2019a])

Let Δ be a fixed finite domain, and let $\mathfrak{I}_\Delta$ be the set of $\mathcal{DL}$-interpretations over Δ. A probability distribution $\mathcal{P}\colon \mathfrak{I}_\Delta \to [0,1]$ is a *probabilistic* $\mathcal{ALC}^{\mathsf{ME}}$-*model* of

1. a general concept inclusion $C \sqsubseteq D$, written $\mathcal{P} \models C \sqsubseteq D$, iff for all $\mathcal{I} \in \mathfrak{I}_\Delta$

$$\mathcal{I} \not\models C \sqsubseteq D \quad \text{implies} \quad \mathcal{P}(\mathcal{I}) = 0,$$

2a. an assertion $C(a)$, written $\mathcal{P} \models C(a)$, iff for all $\mathcal{I} \in \mathfrak{I}_\Delta$

$$\mathcal{I} \not\models C(a) \quad \text{implies} \quad \mathcal{P}(\mathcal{I}) = 0,$$

2b. an assertion $r(a,b)$, written $\mathcal{P} \models r(a,b)$, iff for all $\mathcal{I} \in \mathfrak{I}_\Delta$

$$\mathcal{I} \not\models r(a,b) \quad \text{implies} \quad \mathcal{P}(\mathcal{I}) = 0,$$

3. an $\mathcal{ALC}^{\mathsf{ME}}$-conditional $(D|C)[p]$, written $\mathcal{P} \models (D|C)[p]$, iff

$$\sum_{\mathcal{I} \in \mathfrak{I}_\Delta} |C^{\mathcal{I}}| \cdot \mathcal{P}(\mathcal{I}) > 0 \quad \text{and} \quad \frac{\sum_{\mathcal{I} \in \mathfrak{I}_\Delta} |C^{\mathcal{I}} \cap D^{\mathcal{I}}| \cdot \mathcal{P}(\mathcal{I})}{\sum_{\mathcal{I} \in \mathfrak{I}_\Delta} |C^{\mathcal{I}}| \cdot \mathcal{P}(\mathcal{I})} = p. \tag{7.1}$$

If $\mathcal{P}$ is a probabilistic $\mathcal{ALC}^{\mathsf{ME}}$-model of X, where X is a general concept inclusion, an assertion, or an $\mathcal{ALC}^{\mathsf{ME}}$-conditional, then we write $\mathcal{P} \models X$.

In addition, let $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ be an $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base. If $\mathcal{P}$ is a model of all general concept inclusions, assertions, and $\mathcal{ALC}^{\mathsf{ME}}$-conditionals in $\mathcal{R}$, then $\mathcal{P}$ is a model of $\mathcal{R}$, denoted by $\mathcal{P} \models \mathcal{R}$.

The conditions (1.), (2a.), and (2b.) of Definition 7.1.5 state that general concept inclusions and assertions are treated as hard constraints resp. as facts. That is, when considering a probabilistic model $\mathcal{P}$ of an $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$, only $\mathcal{DL}$-interpretations from $\mathfrak{I}_\Delta$ in which all general concept inclusions and all assertions from $\mathcal{R}$ hold possibly have a positive probability. If an interpretation does not satisfy any general concept inclusion or any assertion, then it is immediately assigned the probability 0 instead. Consequently, the probabilities of the $\mathcal{DL}$-interpretations $\mathcal{I} \in \mathfrak{I}_\Delta$ which satisfy all general concept inclusions, denoted by $\mathcal{I} \models \mathcal{T}$, and all assertions, $\mathcal{I} \models \mathcal{A}$, sum up to 1,

$$\sum_{\mathcal{I} \in \mathfrak{I}_\Delta^\mathcal{R}} \mathcal{P}(\mathcal{I}) = 1, \quad \text{where} \quad \mathfrak{I}_\Delta^\mathcal{R} = \{\mathcal{I} \in \mathfrak{I}_\Delta \mid \mathcal{I} \models \mathcal{T} \quad \text{and} \quad \mathcal{I} \models \mathcal{A}\}.$$

The condition (3.) of Definition 7.1.5 is the *aggregating semantics* (cf. Definition 3.4.1) formulated for $\mathcal{ALC}^{\mathsf{ME}}$-conditionals. In this formulation, the aggregating semantics captures the concept of conditional probabilities by weighting the probabilities $\mathcal{P}(\mathcal{I})$ with the number of individuals for which the conditional $(D|C)[p]$ is *applicable* ($|C^\mathcal{I}|$) respectively *verified* ($|C^\mathcal{I} \cap D^\mathcal{I}|$) in $\mathcal{I}$. In the context of $\mathcal{ALC}^{\mathsf{ME}}$, it becomes very clear that the aggregating semantics mimics statistical probabilities from a subjective point of view, and that probabilities can be understood as degrees of belief in accordance with type 2 probabilities following the classification of Halpern [Halpern, 1990]. We sharpen the intuition by discussing the extreme cases from Section 3.4.

If $\mathcal{P}_\mathcal{I}$ is a Dirac distribution on $\mathfrak{I}_\Delta^\mathcal{R}$ assigning the probability $\mathcal{P}_\mathcal{I}(\mathcal{I}) = 1$ to a distinct interpretation $\mathcal{I} \in \mathfrak{I}_\Delta^\mathcal{R}$, which means that a reasoner with belief state $\mathcal{P}_\mathcal{I}$ is certain that $\mathcal{I}$ reflects the real world (best), then the aggregating semantics reduces to counting the relative frequency of the individuals which have both properties C and D in the interpretation $\mathcal{I}$ compared to the individuals which have property C:

$$\mathcal{P}_I \models (D|C)[p] \quad \text{iff} \quad |C^\mathcal{I}| > 0 \quad \text{and} \quad \frac{|C^\mathcal{I} \cap D^\mathcal{I}|}{|C^\mathcal{I}|} = p.$$

If $\mathcal{P}_\mathcal{R}^u$ is the uniform distribution on $\mathfrak{I}_\Delta^\mathcal{R}$, which means that a reasoner with belief state $\mathcal{P}_\mathcal{R}^u$ is maximally unconfident with her beliefs, then the aggregating semantics means counting the same relative frequency but spread over all interpretations. That is, the numbers of individuals which have properties C and D are summed up over all interpretations $\mathcal{I} \in \mathfrak{I}_\Delta^\mathcal{R}$ and divided through the respective sum of the numbers of individuals which have property C:

$$\mathcal{P}_\mathcal{R}^u \models (D|C)[p] \quad \text{iff} \quad \sum_{\mathcal{I} \in \mathfrak{I}_\Delta^\mathcal{R}} |C^\mathcal{I}| > 0 \quad \text{and} \quad \frac{\sum_{\mathcal{I} \in \mathfrak{I}_\Delta^\mathcal{R}} |C^\mathcal{I} \cap D^\mathcal{I}|}{\sum_{\mathcal{I} \in \mathfrak{I}_\Delta^\mathcal{R}} |C^\mathcal{I}|} = p.$$

This amounts to calculating the statistical frequency of D given C.

If an $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base has a probabilistic model, then it is called *consistent*. Note that the consistency of an $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base depends on the domain size $|\Delta|$ (cf. also Example 4.1.8). Probabilistic models yield a nonmonotonic inference relation (cf. Definition 4.1.10).

Definition 7.1.6: Probabilistic Inference in $\mathcal{ALC}^{\mathsf{ME}}$

(based on [Wilhelm et al., 2019a])

Let $\mathcal{R}$ be a consistent $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base, and let $\mathcal{P}$ be a probabilistic model of $\mathcal{R}$. Then, we define

$$\mathcal{R} \models_{\mathcal{P}} \begin{cases} C(a) & \text{iff} \quad \mathcal{P} \models C(a) \\ r(a,b) & \text{iff} \quad \mathcal{P} \models r(a,b) \\ C \sqsubseteq D & \text{iff} \quad \mathcal{P} \models C \sqsubseteq D, \\ (D|C)[p] & \text{iff} \quad \mathcal{P} \models (D|C)[p] \end{cases} \tag{7.2}$$

where $a, b \in \mathcal{N}_I$, $r \in \mathcal{N}_R$, C and D are $\mathcal{ALC}^{\mathsf{ME}}$-concepts, and p is a probability.

As a straightforward generalization of Definition 7.1.5, we assign probabilities to general concept inclusions and to assertions as well.

Definition 7.1.7: Probabilistic Interpretation of $\mathcal{ALC}^{\mathsf{ME}}$-Axioms

Let Δ be a fixed finite domain, let $\mathfrak{I}_\Delta$ be the set of $\mathcal{DL}$-interpretations over Δ, and let $\mathcal{P}\colon \mathfrak{I}_\Delta \to [0,1]$ be a probability distribution over $\mathfrak{I}_\Delta$. For general concept inclusions $C \sqsubseteq D$ and assertions $C(a)$ resp. $r(a,b)$, we define

$$\begin{aligned} \mathcal{P}(C \sqsubseteq D) &= \sum_{\mathcal{I} \in \mathfrak{I}_\Delta \colon C^{\mathcal{I}} \subseteq D^{\mathcal{I}}} \mathcal{P}(\mathcal{I}), \\ \mathcal{P}(C(a)) &= \sum_{\mathcal{I} \in \mathfrak{I}_\Delta \colon a^{\mathcal{I}} \in C^{\mathcal{I}}} \mathcal{P}(\mathcal{I}), \\ \mathcal{P}(r(a,b)) &= \sum_{\mathcal{I} \in \mathfrak{I}_\Delta \colon (a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}} \mathcal{P}(\mathcal{I}). \end{aligned}$$

The probabilistic interpretation of $\mathcal{ALC}^{\mathsf{ME}}$-axioms given by Definition 7.1.7 is a proper generalization of Definition 7.1.5 in the sense that

$$\begin{aligned} &\mathcal{P} \models C \sqsubseteq D && \text{iff} \quad \mathcal{P}(C \sqsubseteq D) = 1, \\ &\mathcal{P} \models C(a) && \text{iff} \quad \mathcal{P}(C(a)) = 1, \\ &\mathcal{P} \models r(a,b) && \text{iff} \quad \mathcal{P}(r(a,b)) = 1. \end{aligned}$$

For instance, one has $\mathcal{P}(C \sqsubseteq D) = 1$ iff $\sum_{\mathcal{I} \in \mathfrak{I}_\Delta \colon C^{\mathcal{I}} \subseteq D^{\mathcal{I}}} \mathcal{P}(\mathcal{I}) = 1$ which holds iff $\mathcal{P}(\mathcal{I}) = 0$ for all $\mathcal{I} \in \mathfrak{I}_\Delta$ with $\mathcal{I} \not\models C \sqsubseteq D$ which holds iff $\mathcal{P} \models C \sqsubseteq D$ by definition.

▶ Principle of Maximum Entropy and $\mathcal{ALC}^{\mathsf{ME}}$

Like relational knowledge bases in $\mathfrak{R}(\Sigma)$, consistent $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases usually have infinitely many probabilistic models. For the same reasons as discussed in Section 4.2, we select the model of $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases which maximizes the entropy in order to draw inferences [Shannon, 1949; Jaynes, 1957a,b; Paris and Vencovská, 1990]. In particular, from a commonsense point of view, the *maximum entropy model* is the most appropriate choice for the model selection task [Paris, 1998].

Definition 7.1.8: MaxEnt Model of $\mathcal{ALC}^{\mathsf{ME}}$-Knowledge Bases

[Wilhelm et al., 2019a]

Let Δ be a fixed finite domain, let $\mathfrak{I}_\Delta$ be the set of $\mathcal{DL}$-interpretations over Δ, and let $\mathcal{R}$ be a $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base which is consistent wrt. Δ. The *maximum entropy model* $\mathcal{P}^{\mathsf{ME}}_{\Delta,\mathcal{R}}$ of $\mathcal{R}$ wrt. Δ is the unique probability distribution over the set of $\mathcal{ALC}^{\mathsf{ME}}$-interpretations $\mathfrak{I}_\Delta$ which models $\mathcal{R}$ and maximizes the entropy. That is,

$$\mathcal{P}^{\mathsf{ME}}_{\Delta,\mathcal{R}} = \arg\max_{\mathcal{P}\models\mathcal{R}} - \sum_{\mathcal{I}\in\mathfrak{I}_\Delta} \mathcal{P}(\mathcal{I}) \cdot \log \mathcal{P}(\mathcal{I}) \tag{7.3}$$

where the convention $0 \cdot \log 0 = 0$ applies.

The existence and the uniqueness of $\mathcal{P}^{\mathsf{ME}}_{\Delta,\mathcal{R}}$ for consistent $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases $\mathcal{R}$ follow for the same reasons as in the relational case (cf. Section 4.2).

Recall that the maximum entropy model is the solution of a nonlinear optimization problem and can be calculated only approximately in general. This is typically done by solving the dual optimization problem (cf. [Boyd and Vandenberghe, 2009] and Section 4.3), which leads, in the case of $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases, to the product representation (cf. Proposition 4.3.2)

$$\mathcal{P}^{\mathsf{ME}}_{\Delta,\mathcal{R}}(\mathcal{I}) = \begin{cases} \alpha_0 \cdot \prod_{i=1}^{n} \alpha_i^{f_i(\mathcal{I})} & \text{if} \quad \mathcal{I} \in \mathfrak{I}^{\mathcal{R}}_\Delta \\ 0 & \text{if} \quad \mathcal{I} \in \mathfrak{I}_\Delta \setminus \mathfrak{I}^{\mathcal{R}}_\Delta \end{cases}, \tag{7.4}$$

where, for $i = 1, \ldots, n$ when $n = |\mathcal{C}|$, the index i refers to the i-th conditional $(D_i|C_i)[p_i]$ in $\mathcal{C}$,

$$f_i(\mathcal{I}) = |C_i^{\mathcal{I}} \cap D_i^{\mathcal{I}}| - p_i \cdot |C_i^{\mathcal{I}}|, \qquad \mathcal{I} \in \mathfrak{I}_\Delta, \tag{7.5}$$

is the *feature function* wrt. the i-th conditional in $\mathcal{C}$ (cf. Definition 3.4.3), $\vec{\alpha}^{\mathsf{ME}}_{\Delta,\mathcal{R}} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n_{>0}$ is a ME-vector (cf. Definition 4.3.1) which is given as a solution of the maximum entropy equation system (cf. Definition 4.3.7)

$$\sum_{\mathcal{I}\in\mathfrak{I}_{\Delta}^{\mathcal{R}}} f_i(\mathcal{I}) \cdot \prod_{j=1}^{n} \alpha_j^{f_j(\mathcal{I})} = 0, \qquad i = 1, \ldots, n, \tag{7.6}$$

and where

$$\alpha_0 = \Big(\sum_{\mathcal{I}\in\mathfrak{I}_{\Delta}^{\mathcal{R}}} \prod_{i=1}^{n} \alpha_i^{f_i(\mathcal{I})} \Big)^{-1} \tag{7.7}$$

is a normalizing constant ensuring that the probabilities $\mathcal{P}^{\mathsf{ME}}_{\Delta,\mathcal{R}}(\mathcal{I})$ sum up to 1. Recall that the product representation (7.6) of $\mathcal{P}^{\mathsf{ME}}_{\Delta,\mathcal{R}}$ requires the p-consistency of $\mathcal{R}$ wrt. Δ, i.e., the positivity of $\mathcal{P}^{\mathsf{ME}}_{\Delta,\mathcal{R}}$ on $\mathfrak{I}_{\Delta}^{\mathcal{R}}$ (cf. Definition 4.2.5 and Definition 4.2.6).

In analogy to (5.7), we can define the probability distribution

$$\mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}(\mathcal{I}) = \begin{cases} \alpha_0^* \cdot \prod_{i=1}^{n} (\alpha_i^*)^{f_i(\mathcal{I})} & \mathcal{I} \in \mathfrak{I}_{\Delta}^{\mathcal{R}} \\ 0 & \mathcal{I} \in \mathfrak{I}_{\Delta} \setminus \mathfrak{I}_{\Delta}^{\mathcal{R}} \end{cases} \tag{7.8}$$

with respect to an approximation $\vec{\alpha}^* \in \mathbb{Q}^n_{>0}$ to $\vec{\alpha}^{\mathsf{ME}}_{\mathcal{R}}$ and the normalization constant α_0^* as an approximation to $\mathcal{P}^{\mathsf{ME}}_{\Delta,\mathcal{R}}$. Note that $\mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}$ is an exact model of $\mathcal{T}$ and $\mathcal{A}$ by construction, and models $\mathcal{C}$ up to a deviation depending on the precision of the approximation $\vec{\alpha}^*$.

Instead of discussing the principle of maximum entropy for $\mathcal{ALC}^{\mathsf{ME}}$ in further detail, we prove that $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases can be compiled into knowledge bases in $\mathfrak{R}(\Sigma)$ so that all results from relational maximum entropy reasoning immediately follow for $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases, too.

▶ $\mathcal{ALC}^{\mathsf{ME}}$ as a Fragment of $\mathcal{RPCL}(\Sigma)$

We prove that the probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ is a fragment of $\mathcal{RPCL}(\Sigma)$.[1] With “fragment” we mean that there is a translation of all $\mathcal{ALC}^{\mathsf{ME}}$-expressions to appropriate expressions in $\mathcal{RPCL}(\Sigma)$, or $\mathcal{RL}^S(\Sigma)$ in case of $\mathcal{ALC}^{\mathsf{ME}}$-axioms, which are interpreted in the same way as the $\mathcal{ALC}^{\mathsf{ME}}$-expressions. Because we have already seen that the Description Logic $\mathcal{ALC}^{\mathsf{ffd}}$, i.e., $\mathcal{ALC}$ with a fixed finite domain and the unique name assumption, is a fragment of $\mathcal{RL}(\Sigma)$ (cf. Definition 2.5.1), we can take the translation from Definition 2.5.1 up and just have to translate the $\mathcal{ALC}^{\mathsf{ME}}$-conditionals which are newly introduced in $\mathcal{ALC}^{\mathsf{ME}}$. Having said that, we recall the basics of the translation from Definition 2.5.1 for the avoidance of doubt nonetheless.

---

[1] To be more precise, $\mathcal{ALC}^{\mathsf{ME}}$-conditionals refer to conditionals in $\mathcal{RPCL}(\Sigma)$ and $\mathcal{ALC}^{\mathsf{ME}}$-axioms refer to relational sentences in $\mathcal{RL}^S(\Sigma)$.

Like in Definition 2.5.1, we consider the relational signature

$$\Sigma(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R) = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$$

given by (cf. (2.2))

$$\mathsf{Const}_\Sigma = \mathcal{N}_I \quad \text{and} \quad \mathsf{Pred}_\Sigma = \{P_A/1 \mid A \in \mathcal{N}_C\} \cup \{R_r/2 \mid r \in \mathcal{N}_R\},$$

and we recall the mapping $\gamma_{\mathfrak{C}}$ from Definition 2.5.1 which translates $\mathcal{ALC}^{\mathsf{ffd}}$-concepts to formulas in $\mathcal{RL}(\Sigma)$. We extend the $\mathcal{ALC}^{\mathsf{ffd}}$-translation γ from Definition 2.5.1, which builds upon $\gamma_{\mathfrak{C}}$, to conditionals as follows.

Definition 7.1.9: Translation from $\mathcal{ALC}^{\mathsf{ME}}$ to $\mathcal{RPCL}(\Sigma)$

(based on [Rudolph, 2011])

Let $\mathcal{N}_I$, $\mathcal{N}_C$, and $\mathcal{N}_R$ be the sets of individual, concept, and role names of the Description Logic $\mathcal{ALC}^{\mathsf{ME}}$. Further, let $\mathfrak{C}$ be the set of $\mathcal{ALC}^{\mathsf{ME}}$-concepts over $\mathcal{N}_R$ and $\mathcal{N}_C$, let Vars be a set of logical variables, and let $\Sigma(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R)$ be the first-order signature induced by $\mathcal{N}_I$, $\mathcal{N}_C$, and $\mathcal{N}_R$. Then, we inductively define a mapping $\gamma_{\mathfrak{C}} \colon \mathfrak{C} \times \mathsf{Vars} \to \mathcal{RPCL}(\Sigma)$ in analogy to Definition 2.5.1 which translates $\mathcal{ALC}^{\mathsf{ME}}$-concepts to formulas in $\mathcal{RPCL}(\Sigma)$ by

$$\begin{aligned}
\gamma_{\mathfrak{C}}(\top, X) &= \top, \\
\gamma_{\mathfrak{C}}(\bot, X) &= \bot, \\
\gamma_{\mathfrak{C}}(A, X) &= P_A(X), \\
\gamma_{\mathfrak{C}}(\neg C, X) &= \neg\gamma_{\mathfrak{C}}(C, X), \\
\gamma_{\mathfrak{C}}(C \sqcap D, X) &= \gamma_{\mathfrak{C}}(C, X) \wedge \gamma_{\mathfrak{C}}(D, X), \\
\gamma_{\mathfrak{C}}(C \sqcup D, X) &= \gamma_{\mathfrak{C}}(C, X) \vee \gamma_{\mathfrak{C}}(D, X), \\
\gamma_{\mathfrak{C}}(\exists r.C, X) &= \exists Y.(R_r(X, Y) \wedge \gamma_{\mathfrak{C}}(C, Y)), & Y \neq X, \\
\gamma_{\mathfrak{C}}(\forall r.C, X) &= \forall Y.(R_r(X, Y) \Rightarrow \gamma_{\mathfrak{C}}(C, Y)), & Y \neq X,
\end{aligned}$$

where $A \in \mathcal{N}_C$ is a concept name, $C, D \in \mathfrak{C}$ are $\mathcal{ALC}^{\mathsf{ME}}$-concepts, $r \in \mathcal{N}_R$ is a role name, and $X, Y \in \mathsf{Vars}$ are variables. Based on this mapping, we further define the $\mathcal{ALC}^{\mathsf{ME}}$*-translation* $\gamma \colon \mathfrak{A} \to \mathcal{RPCL}(\Sigma)$, where $\mathfrak{A}$ is the set of all general concept inclusions, assertions, and conditionals in $\mathcal{ALC}^{\mathsf{ME}}$, by

$$\begin{aligned}
\gamma(C \sqsubseteq D) &= \forall X.(\gamma_{\mathfrak{C}}(C, X) \Rightarrow \gamma_{\mathfrak{C}}(D, X)) \\
\gamma(C(a)) &= \gamma_{\mathfrak{C}}(C, X)\langle X/a\rangle, \\
\gamma(r(a, b)) &= R_r(a, b),
\end{aligned}$$

$$\gamma((D|C)[p]) = (\gamma_{\mathfrak{C}}(D, X)|\gamma_{\mathfrak{C}}(C, X))[p].$$

The translation γ can be extended to a TBox $\mathcal{T}$, an ABox $\mathcal{A}$, resp. a CBox $\mathcal{C}$ by

$$\begin{aligned}
\gamma(\mathcal{T}) &= \bigwedge_{(C \sqsubseteq D) \in \mathcal{T}} \gamma(C \sqsubseteq D),\\
\gamma(\mathcal{A}) &= \bigwedge_{C(a) \in \mathcal{A}} \gamma(C(a)) \wedge \bigwedge_{r(a,b) \in \mathcal{A}} R_r(a, b),\\
\gamma(\mathcal{C}) &= \{\gamma((D|C)[p]) \mid (D|C)[p] \in \mathcal{C}\}.
\end{aligned}$$

Finally, the translation of an $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base $(\mathcal{T}, \mathcal{A}, \mathcal{C})$ into a relational knowledge base $\mathcal{R} = (\mathcal{F}_{\mathcal{R}}, \mathcal{B}_{\mathcal{R}})$ is given by

$$\mathcal{F}_{\mathcal{R}} = \{\gamma(\mathcal{T}), \gamma(\mathcal{A})\}, \quad \text{and} \quad \mathcal{B}_{\mathcal{R}} = \gamma(\mathcal{C}).$$

Note that the translation γ from Definition 7.1.9 is the same as the translation from Definition 2.5.1 except for its capability of translating (sets of) $\mathcal{ALC}^{\mathsf{ME}}$-conditionals to (sets of) relational probabilistic conditionals in $\mathcal{RPCL}(\Sigma)$. The translation of $\mathcal{ALC}^{\mathsf{ME}}$-conditionals results in conditionals in $\mathcal{RPCL}(\Sigma)$ with one free variable X. Further note that the translation of a TBox $\mathcal{T}$ resp. an ABox $\mathcal{A}$ as given in Definition 7.1.9 is semantically equivalent to

$$\begin{aligned}
\gamma(\mathcal{T}) &= \{\gamma(C \sqsubseteq D) \mid C \sqsubseteq D \in \mathcal{T}\},\\
\gamma(\mathcal{A}) &= \{\gamma(C(a)) \mid C(a) \in \mathcal{A}\} \cup \{\gamma(r(a, b)) \mid r(a, b) \in \mathcal{A}\}.
\end{aligned}$$

It remains to show that the probabilistic interpretation of $\mathcal{ALC}^{\mathsf{ME}}$-expressions is identical with the probabilistic interpretation of the respective translations to $\mathcal{RPCL}(\Sigma)$. For this, let Δ be the fixed finite domain of $\mathcal{ALC}^{\mathsf{ME}}$ so that $\mathfrak{I}_\Delta = \mathfrak{I}_\Delta(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R)$ is the set of $\mathcal{DL}$-interpretations in $\mathcal{ALC}^{\mathsf{ME}}$. Every interpretation $\mathcal{I} \in \mathfrak{I}_\Delta$ can be uniquely identified with a possible world $\omega \in \Omega(\Sigma(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R))$ via

$$\begin{aligned}
\mathcal{I} \sim \omega \quad \text{iff} \quad \omega = {} & \bigwedge_{A \in \mathcal{N}_C} \left(\bigwedge_{a \in \mathcal{N}_I :\, a^{\mathcal{I}} \in A^{\mathcal{I}}} P_A(a) \wedge \bigwedge_{a \in \mathcal{N}_I :\, a^{\mathcal{I}} \notin A^{\mathcal{I}}} \overline{P_A(a)} \right)\\
& \bigwedge_{r \in \mathcal{N}_R} \left(\bigwedge_{(a,b) \in \mathcal{N}_I^2 :\, (a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}} R_r(a, b) \wedge \bigwedge_{(a,b) \in \mathcal{N}_I^2 :\, (a^{\mathcal{I}}, b^{\mathcal{I}}) \notin r^{\mathcal{I}}} \overline{R_r(a, b)} \right).
\end{aligned}$$

We denote the possible world ω which is in one-to-one-correspondence with th $\mathcal{DL}$-interpretation $\mathcal{I}$ with $\omega_{\mathcal{I}}$. Then, $\Omega(\Sigma(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R)) = \{\omega_{\mathcal{I}} \mid \mathcal{I} \in \mathfrak{I}_\Delta\}$ holds, and probability distributions over $\Omega(\Sigma(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R))$ are in a one-to-one correspondence with probability distributions over $\mathfrak{I}_\Delta$. We write $\mathcal{P}_\Omega$ for the probability distribution over $\Omega(\Sigma(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R))$ which is induced by the probability distribution $\mathcal{P}$ over $\mathfrak{I}_\Delta$.

Proposition 7.1.10: Correctness of the Translation γ

Let Δ be the fixed finite domain of $\mathcal{ALC}^{\mathsf{ME}}$ so that $\mathfrak{I}_\Delta = \mathfrak{I}_\Delta(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R)$ is the set of $\mathcal{DL}$-interpretations in $\mathcal{ALC}^{\mathsf{ME}}$. Further, let $\Sigma(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R)$ be the first-order signature induced by $\mathcal{N}_I$, $\mathcal{N}_C$, and $\mathcal{N}_R$, let $\mathcal{P}\colon \mathfrak{I}_\Delta \to [0,1]$ be a probability distribution over $\mathfrak{I}_\Delta$, and let $\mathcal{P}_\Omega\colon \Omega(\Sigma) \to [0,1]$ be the corresponding probability distribution over $\Omega(\Sigma(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R))$. Then,

$$\begin{aligned}
\mathcal{P} &\models C(a) && \text{iff} \quad \mathcal{P}_\Omega \models \gamma(C(a)),\\
\mathcal{P} &\models r(a,b) && \text{iff} \quad \mathcal{P}_\Omega \models \gamma(r(a,b)),\\
\mathcal{P} &\models C \sqsubseteq D && \text{iff} \quad \mathcal{P}_\Omega \models \gamma(C \sqsubseteq D),\\
\mathcal{P} &\models (D|C)[p] && \text{iff} \quad \mathcal{P}_\Omega \models \gamma((D|C)[p]),
\end{aligned}$$

where $a, b \in \mathcal{N}_I$, C, D are $\mathcal{ALC}^{\mathsf{ME}}$-concepts, $r \in \mathcal{N}_R$, and $p \in [0,1]$. Consequently, for an $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base $(\mathcal{T}, \mathcal{A}, \mathcal{C})$ we have

$$\mathcal{P} \models (\mathcal{T}, \mathcal{A}, \mathcal{C}) \quad \text{iff} \quad \mathcal{P}_\Omega \models (\{\gamma(\mathcal{T}), \gamma(\mathcal{A})\}, \gamma(\mathcal{C}))$$

Proof. We abbreviate $\Sigma = \Sigma(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R)$. Then,

$$\begin{aligned}
\mathcal{P}(C(a)) &= \sum_{\mathcal{I} \in \mathfrak{I}_\Delta \colon a^{\mathcal{I}} \in C^{\mathcal{I}}} \mathcal{P}(\mathcal{I}) &&= \sum_{\omega_{\mathcal{I}} \colon a^{\mathcal{I}} \in C^{\mathcal{I}}} \mathcal{P}_\Omega(\omega_{\mathcal{I}})\\
&= \sum_{\omega \in \Omega(\Sigma) \colon \omega \models C(a)} \mathcal{P}_\Omega(\omega) &&= \sum_{\omega \in \Omega(\Sigma) \colon \omega \models \gamma(C(a)} \mathcal{P}_\Omega(\omega)\\
&= \mathcal{P}_\Omega(\gamma(C(a))),
\end{aligned}$$

from which $\mathcal{P} \models C(a)$ iff $\mathcal{P}_\Omega \models \gamma(C(a))$ follows. Analogously, we have

$$\begin{aligned}
\mathcal{P}(r(a,b)) &= \sum_{\mathcal{I} \in \mathfrak{I}_\Delta \colon (a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}} \mathcal{P}(\mathcal{I}) &&= \sum_{\omega_{\mathcal{I}} \colon (a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}} \mathcal{P}_\Omega(\omega_{\mathcal{I}})\\
&= \sum_{\omega \in \Omega(\Sigma) \colon \omega \models r(a,b)} \mathcal{P}_\Omega(\omega) &&= \sum_{\omega \in \Omega(\Sigma) \colon \omega \models \gamma(r(a,b)} \mathcal{P}_\Omega(\omega)\\
&= \mathcal{P}_\Omega(\gamma(r(a,b))),
\end{aligned}$$

and

$$\mathcal{P}(C \sqsubseteq D) = \sum_{\mathcal{I} \in \mathfrak{I}_\Delta \colon C^{\mathcal{I}} \subseteq D^{\mathcal{I}}} \mathcal{P}(\mathcal{I}) \qquad = \sum_{\omega_{\mathcal{I}} \colon C^{\mathcal{I}} \subseteq D^{\mathcal{I}}} \mathcal{P}_\Omega(\omega_{\mathcal{I}})$$

$$
\begin{aligned}
&= \sum_{\omega \in \Omega(\Sigma)\colon \omega \models \forall X.(\gamma_{\mathfrak{C}}(C,X) \Rightarrow \gamma_{\mathfrak{C}}(D,X))} \mathcal{P}_\Omega(\omega) \quad = \sum_{\omega \in \Omega(\Sigma)\colon \omega \models \gamma(C \sqsubseteq D)} \mathcal{P}_\Omega(\omega) \\
&= \mathcal{P}_\Omega(\gamma(C \sqsubseteq D)).
\end{aligned}
$$

Further, for $\mathcal{ALC}^{\mathsf{ME}}$-conditionals, we have

$$
\begin{aligned}
&\mathcal{P} \models (D|C)[p] \\
\text{iff} \quad & \sum_{\mathcal{I} \in \mathfrak{I}_\Delta} |C^{\mathcal{I}}| \cdot \mathcal{P}(\mathcal{I}) > 0 \quad \text{and} \quad \frac{\sum_{\mathcal{I} \in \mathfrak{I}_\Delta} |C^{\mathcal{I}} \cap D^{\mathcal{I}}| \cdot \mathcal{P}(\mathcal{I})}{\sum_{\mathcal{I} \in \mathfrak{I}_\Delta} |C^{\mathcal{I}}| \cdot \mathcal{P}(\mathcal{I})} = p, \\
\text{iff} \quad & \sum_{\omega \in \Omega(\Sigma)} |\{c \in \mathsf{Const}_\Sigma \mid \omega \models \gamma_{\mathfrak{C}}(C,X)\langle X/c \rangle\}| \cdot \mathcal{P}_\Omega(\omega) > 0 \quad \text{and} \\
& \frac{\sum_{\omega \in \Omega(\Sigma)} |\{c \in \mathsf{Const}_\Sigma \mid \omega \models \gamma_{\mathfrak{C}}(C,X)\langle X/c \rangle \wedge \gamma_{\mathfrak{C}}(D,X)\langle X/c \rangle\}| \cdot \mathcal{P}_\Omega(\omega)}{\sum_{\omega \in \Omega(\Sigma)} |\{c \in \mathsf{Const}_\Sigma \mid \omega \models \gamma_{\mathfrak{C}}(C,X)\langle X/c \rangle\}| \cdot \mathcal{P}_\Omega(\omega)} = p \\
\text{iff} \quad & \sum_{c \in \mathsf{Const}_\Sigma} \sum_{\omega \in \Omega(\Sigma)\colon \omega \models \tau_{\mathfrak{C}}(C,X)\langle X/c \rangle} \mathcal{P}_\Omega(\omega) > 0 \quad \text{and} \\
& \frac{\sum_{c \in \mathsf{Const}_\Sigma} \sum_{\omega \in \Omega(\Sigma)\colon \omega \models \tau_{\mathfrak{C}}(C,X)\langle X/c \rangle \wedge \tau_{\mathfrak{C}}(D,X)\langle X/c \rangle} \mathcal{P}_\Omega(\omega)}{\sum_{c \in \mathsf{Const}_\Sigma} \sum_{\omega \in \Omega(\Sigma)\colon \omega \models \tau_{\mathfrak{C}}(C,X)\langle X/c \rangle} \mathcal{P}_\Omega(\omega)} = p \\
\text{iff} \quad & \sum_{c \in \mathsf{Const}_\Sigma} \mathcal{P}_\Omega(\tau_{\mathfrak{C}}(C,X)\langle X/c \rangle) > 0 \quad \text{and} \\
& \frac{\sum_{c \in \mathsf{Const}_\Sigma} \mathcal{P}_\Omega(\tau_{\mathfrak{C}}(C,X)\langle X/c \rangle \wedge \tau_{\mathfrak{C}}(D,X)\langle X/c \rangle)}{\sum_{c \in \mathsf{Const}_\Sigma} \mathcal{P}_\Omega(\tau_{\mathfrak{C}}(C,X)\langle X/c \rangle)} = p \\
\text{iff} \quad & \mathcal{P}_\Omega \models (\gamma_{\mathfrak{C}}(D,X) | \gamma_{\mathfrak{C}}(C,X))[p] \\
\text{iff} \quad & \mathcal{P}_\Omega \models \gamma((D|C)[p]).
\end{aligned}
$$

The statement

$$
\mathcal{P} \models (\mathcal{T}, \mathcal{A}, \mathcal{C}) \quad \text{iff} \quad \mathcal{P}_\Omega \models (\{\gamma(\mathcal{T}), \gamma(\mathcal{A})\}, \gamma(\mathcal{C}))
$$

is a direct consequence of these results then. □

As a result of Proposition 7.1.10, one can draw probabilistic inferences from $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases by translating them to equivalent knowledge bases $\mathcal{R} \in \mathfrak{R}(\Sigma)$ first, and drawing the respective inference from the translation $\mathcal{R}$ thereafter. In particular, one has recourse to the concept of *typed model counting* (Section 6.2) and the CIS algorithm (Section 5.2) in this way. Apart from this possibility, we propose a direct method for drawing lifted inferences in $\mathcal{ALC}^{\mathsf{ME}}$ in the next section.

7.2 Lifted Inference in $\mathcal{ALC}^{\mathsf{ME}}$

In this section, we prove that drawing inferences from an $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base without assertions based on a proper approximation of the maximum entropy model (cf. Definition 7.8) is possible in time polynomial in the domain size $k = |\Delta|$. While typed model counting provides a very general framework for drawing maximum entropy inferences (cf. Section 6.4), we apply counting techniques which are highly adapted to $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases here and, therewith, optimize the heavy machinery of typed model counting to this case. An important aspect of our deliberations is that we can partition Δ based on the notion of *$\mathcal{DL}$-types*. If this is not possible, for instance in more expressive Description Logics involving role inclusions, full typed model counting can be applied anyway, but then without the guarantee of domain-lifted inferences. This section is based on [Wilhelm et al., 2019a].

▶ Problem Setting

We assume that a consistent $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ without assertions is given, i.e., $\mathcal{A} = \emptyset$. We set $n = |\mathcal{C}|$ and enumerate the conditionals in $\mathcal{C}$ and refer to the i-th conditional in $\mathcal{C}$ by $(D_i|C_i)[p_i]$. In addition, we impose the following requirements on $\mathcal{R}$:

- We assume that the $\mathcal{ALC}^{\mathsf{ME}}$-concepts which are mentioned in $\mathcal{R}$ are built using the concept constructors negation ($\neg C$), conjunction ($C \sqcap D$), and existential restriction ($\exists r.A$) with A being a concept name only. Also, we disallow double negation. Instead, whenever the negation of an already negated concept is mentioned, we mean the concept itself. Note that this is without loss of generality. For instance, if one wants to express an existential restriction $\exists r.C$ where C is an arbitrary $\mathcal{ALC}^{\mathsf{ME}}$-concept, one can simply introduce a fresh concept name A, replace C in the existential restriction by A and add the general concept inclusions $A \sqsubseteq C$ and $C \sqsubseteq A$ ("A is defined as C") to the TBox of the knowledge base.
- In general, the probabilities in $\mathcal{ALC}^{\mathsf{ME}}$-conditionals are real numbers. For computational issues, we assume that the probabilities are rational numbers here. That is, for $i = 1, \dots, n$, the probability p_i of the i-th conditional in $\mathcal{C}$ is of the form $p_i = \frac{s_i}{t_i}$ where s_i and t_i are natural numbers satisfying $0 < s_i < t_i$.
- The general remarks on $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases from the beginning of Section 7.1 hold.

In order to be able to refer to the signature of the knowledge base $\mathcal{R}$, we denote the set of all concept names that are mentioned in $\mathcal{R}$ with $\mathsf{sig}_C(\mathcal{R})$, and the set of all role names that are mentioned in $\mathcal{R}$ with $\mathsf{sig}_R(\mathcal{R})$.

Further, we assume that an approximation $\mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}$ of the maximum entropy model $\mathcal{P}^{\mathsf{ME}}_{\Delta,\mathcal{R}}$ of $\mathcal{R}$ is given which is of the form (cf. (7.8))

$$\mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}(\mathcal{I}) = \begin{cases} \alpha_0^* \cdot \prod_{i=1}^n (\alpha_i^*)^{f_i(\mathcal{I})} & \mathcal{I} \in \mathfrak{I}_\Delta^{\mathcal{R}} \\ 0 & \mathcal{I} \in \mathfrak{I}_\Delta \setminus \mathfrak{I}_\Delta^{\mathcal{R}} \end{cases}. \tag{7.9}$$

Our goal is to compute the probability $p \in [0,1]$ for which $\mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}} \models (D|C)[p]$ holds, where C and D are $\mathcal{ALC}^{\mathsf{ME}}$-concepts that also satisfy the above-mentioned requirements. We prove that this is possible in time polynomial in $k = |\Delta|$.

For our purpose, we capture the fact that $\mathcal{DL}$-interpretations with the same conditional structure (cf. Definition 3.2.1) have the same impact on the maximum entropy model and also on its approximation $\mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}$. Here, we refine the notion of conditional structures of $\mathcal{DL}$-interpretations to *conditional impacts* of *$\mathcal{DL}$-types* [Pratt, 1979; Rudolph et al., 2012; Baader and Ecke, 2017] which enables the use of efficient counting strategies. In the context of $\mathcal{ALC}^{\mathsf{ME}}$, the *conditional structure* of a $\mathcal{DL}$-interpretation $\mathcal{I}$ with respect to an $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ is given by

$$\sigma_{\mathcal{R}}(\mathcal{I}) = \prod_{i=1}^{n} (\mathbf{a}_{\mathbf{i}}^{+})^{|C_i^{\mathcal{I}} \cap D_i^{\mathcal{I}}|} \cdot (\mathbf{a}_{\mathbf{i}}^{-})^{|C_i^{\mathcal{I}} \cap (\neg D_i)^{\mathcal{I}}|} \tag{7.10}$$

and reflects how often the conditionals in $\mathcal{C}$ are verified and falsified in $\mathcal{I}$.

▶ Grouping Individuals to $\mathcal{DL}$-Types

Types in Description Logics [Pratt, 1979; Rudolph et al., 2012; Baader and Ecke, 2017] are characterizations of individuals through the concepts they belong to. Here, we use the notion of $\mathcal{DL}$-types in order to refine conditional structures.

Definition 7.2.1: $\mathcal{DL}$-Type *(based on [Baader and Ecke, 2017])*

Let $\mathfrak{C}$ be a set of $\mathcal{ALC}^{\mathsf{ME}}$-concepts which is closed under negation, i.e., for every $\mathcal{ALC}^{\mathsf{ME}}$-concept $C \in \mathfrak{C}$ its negation $\neg C$ is also in $\mathfrak{C}$, where double negations are removed. A subset Γ of $\mathfrak{C}$ is a *$\mathcal{DL}$-type* for $\mathfrak{C}$ iff

- for every $C \in \mathfrak{C}$, either C or $\neg C$ belongs to Γ,
- for every $C \sqcap D \in \mathfrak{C}$ it holds that $C \sqcap D \in \Gamma$ iff $C, D \in \Gamma$.

The set of all $\mathcal{DL}$-types for $\mathfrak{C}$ is denoted by $\mathfrak{T}(\mathfrak{C})$.

We aim at $\mathcal{DL}$-types that are defined with respect to the $\mathcal{ALC}^{\mathsf{ME}}$-concepts which occur in a $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ with $\mathcal{A} = \emptyset$. Let $\mathfrak{C}_{\mathcal{R}}^{+}$ be the set of

Γ	$\rho_{\mathcal{R}_W}(\Gamma)$
$\Gamma_1 = \{ S, W, G, \exists p.G, \neg(\neg S \sqcap \neg\exists p.G)\}$	$\mathbf{a_1^- a_2^-}$
$\Gamma_2 = \{ S, W, G, \neg\exists p.G, \neg(\neg S \sqcap \neg\exists p.G)\}$	$\mathbf{a_1^- a_2^-}$
$\Gamma_3 = \{ S, W, \neg G, \exists p.G, \neg(\neg S \sqcap \neg\exists p.G)\}$	$\mathbf{a_1^- a_2^+}$
$\Gamma_4 = \{ S, W, \neg G, \neg\exists p.G, \neg(\neg S \sqcap \neg\exists p.G)\}$	$\mathbf{a_1^- a_2^+}$
$\Gamma_5 = \{ S, \neg W, G, \exists p.G, \neg(\neg S \sqcap \neg\exists p.G)\}$	$\mathbf{1}$
$\Gamma_6 = \{ S, \neg W, G, \neg\exists p.G, \neg(\neg S \sqcap \neg\exists p.G)\}$	$\mathbf{1}$
$\Gamma_7 = \{ S, \neg W, \neg G, \exists p.G, \neg(\neg S \sqcap \neg\exists p.G)\}$	$\mathbf{1}$
$\Gamma_8 = \{ S, \neg W, \neg G, \neg\exists p.G, \neg(\neg S \sqcap \neg\exists p.G)\}$	$\mathbf{1}$
$\Gamma_9 = \{\neg S, W, G, \exists p.G, \neg(\neg S \sqcap \neg\exists p.G)\}$	$\mathbf{a_1^- a_2^-}$
$\Gamma_{10} = \{\neg S, W, G, \neg\exists p.G, \neg S \sqcap \neg\exists p.G \}$	$\mathbf{a_1^+ a_2^-}$
$\Gamma_{11} = \{\neg S, W, \neg G, \exists p.G, \neg(\neg S \sqcap \neg\exists p.G)\}$	$\mathbf{a_1^- a_2^+}$
$\Gamma_{12} = \{\neg S, W, \neg G, \neg\exists p.G, \neg S \sqcap \neg\exists p.G \}$	$\mathbf{a_1^+ a_2^+}$
$\Gamma_{13} = \{\neg S, \neg W, G, \exists p.G, \neg(\neg S \sqcap \neg\exists p.G)\}$	$\mathbf{1}$
$\Gamma_{14} = \{\neg S, \neg W, G, \neg\exists p.G, \neg S \sqcap \neg\exists p.G \}$	$\mathbf{1}$
$\Gamma_{15} = \{\neg S, \neg W, \neg G, \exists p.G, \neg(\neg S \sqcap \neg\exists p.G)\}$	$\mathbf{1}$
$\Gamma_{16} = \{\neg S, \neg W, \neg G, \neg\exists p.G, \neg S \sqcap \neg\exists p.G \}$	$\mathbf{1}$

Table 7.2: $\mathcal{DL}$-Types in $\mathfrak{T}_{\mathcal{R}_W}$ and their conditional impacts wrt. $\mathcal{R}_W$ from Example 7.2.2. The concept and role names are abbreviated by their first letter.

$\mathcal{ALC}^{\mathsf{ME}}$-concepts and all of their subconcepts which occur in $\mathcal{R}$, i.e.,[2]

$$\mathfrak{C}_{\mathcal{R}}^{+} = \bigcup_{C \sqsubseteq D \in \mathcal{T}} \Big(\mathsf{sub}(C) \cup \mathsf{sub}(D)\Big) \cup \bigcup_{(D|C)[p] \in \mathcal{C}} \Big(\mathsf{sub}(C) \cup \mathsf{sub}(D)\Big).$$

Then, we define its closure under negation with $\mathfrak{C}_{\mathcal{R}} = \mathfrak{C}_{\mathcal{R}}^{+} \cup \{\neg C \mid C \in \mathfrak{C}_{\mathcal{R}}^{+}\}$ and consider the $\mathcal{DL}$-types in $\mathfrak{T}_{\mathcal{R}} = \mathfrak{T}(\mathfrak{C}_{\mathcal{R}})$.

Example 7.2.2: *(cf. [Wilhelm et al., 2019a])*

We consider the $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base $\mathcal{R}_W = (\{F_1\}, \emptyset, \{r_1, r_2\})$ with

$$F_1: \ \mathsf{Generous} \sqsubseteq \mathsf{Wealthy},$$

---

[2] With $\mathsf{sub}(C)$ we denote the set of all subconcepts of an $\mathcal{ALC}^{\mathsf{ME}}$-concept C.

$$\begin{aligned} r_1\colon &\ (\neg\mathsf{Successful} \sqcap \neg\exists\mathsf{patron}.\mathsf{Generous}|\mathsf{Wealthy})[0.1],\\ r_2\colon &\ (\neg\mathsf{Generous}|\mathsf{Wealthy})[0.8], \end{aligned}$$

stating that every person who is generous is wealthy (F_1), every wealthy person most likely (with probability 0.9) is successful in her career or has a generous patron (r_1), and wealthy persons are typically not generous, here with probability 0.8 (r_2). Note that r_1 is equivalent to the conditional

$$(\mathsf{Successful} \sqcup \exists\mathsf{patron}.\mathsf{Generous}|\mathsf{Wealthy})[0.9].$$

The $\mathcal{DL}$-types in $\mathfrak{T}_{\mathcal{R}_W}$ can be computed as follows. First, one extracts all (sub)concepts from $\mathcal{R}_W$ which are

$$\mathfrak{C}^+_{\mathcal{R}_W} = \{\ G,\ \ W,\ \ \neg S \sqcap \neg\exists p.G,\ \ \neg S,\ \ S,\ \ \neg\exists p.G,\ \ \exists p.G,\ \ \neg G\ \},$$

where concept and role names are abbreviated by their first letter. The closure of $\mathfrak{C}^+_{\mathcal{R}_W}$ under negation is

$$\begin{aligned} \mathfrak{C}_{\mathcal{R}_W} = \{ \quad & G, \quad W, \quad S, \quad \exists p.G, \quad \neg S \sqcap \neg\exists p.G,\\ & \neg G,\ \neg W,\ \neg S,\ \neg\exists p.G,\ \neg(\neg S \sqcap \neg\exists p.G) \quad \}. \end{aligned}$$

Then, one builds all possible combinations of the $\mathcal{ALC}^{\mathsf{ME}}$-concepts in $\mathfrak{C}_{\mathcal{R}_W}$ which satisfy the constraints from the definition of $\mathcal{DL}$-types (Definition 7.2.1). Altogether, there are 16 $\mathcal{DL}$-types in $\mathfrak{T}_{\mathcal{R}_W}$ which are listed in Table 7.2.

A $\mathcal{DL}$-type Γ can be identified with the $\mathcal{ALC}^{\mathsf{ME}}$-concept C_Γ that is the conjunction of all $\mathcal{ALC}^{\mathsf{ME}}$-concepts in Γ. If Γ and Γ' are different $\mathcal{DL}$-types, $\Gamma \neq \Gamma'$, then C_Γ and $C_{\Gamma'}$ are disjoint, i.e., $C_\Gamma^{\mathcal{I}} \cap C_{\Gamma'}^{\mathcal{I}} = \emptyset$ holds for all $\mathcal{I} \in \mathfrak{I}_\Delta$. Every $\mathcal{ALC}^{\mathsf{ME}}$-concept $C \in \mathfrak{C}$ can be expressed as a disjunction of such disjoint *type concepts* (cf. [Baader and Ecke, 2017] and also [Wilhelm et al., 2019a]).

Proposition 7.2.3: Partitioning the Domain With $\mathcal{DL}$-Types

(based on [Baader and Ecke, 2017])

Let $\mathfrak{C}$ be a set of $\mathcal{ALC}^{\mathsf{ME}}$-concepts with $D \in \mathfrak{C}$, and let $\mathcal{I} \in \mathfrak{I}_\Delta$ be a $\mathcal{DL}$-interpretation. Then,

$$D \equiv \bigsqcup_{\Gamma \in \mathfrak{T}(\mathfrak{C})\colon D \in \Gamma} C_\Gamma \quad \text{and} \quad |D^{\mathcal{I}}| = \sum_{\Gamma \in \mathfrak{T}(\mathfrak{C})\colon D \in \Gamma} |C_\Gamma^{\mathcal{I}}|. \tag{7.11}$$

Further, the cardinalities $|C_\Gamma^{\mathcal{I}}|$ of all type concepts with $\Gamma \in \mathfrak{T}(\mathfrak{C})$ sum up to the domain size $|\Delta|$:

$$\bigsqcup_{\Gamma \in \mathfrak{T}(\mathfrak{C})} C_\Gamma \equiv \top \quad \text{and} \quad \sum_{\Gamma \in \mathfrak{T}(\mathfrak{C})} |C_\Gamma^{\mathcal{I}}| = |\Delta|. \tag{7.12}$$

Proof. For the proof of (7.11) see [Baader and Ecke, 2017]. In order to prove (7.12), let $\mathcal{I} \in \mathfrak{I}_\Delta$ be a $\mathcal{DL}$-interpretation, and let $c \in \Delta$. We define $\Gamma_c = \{C \in \mathfrak{C} \mid c \in C^{\mathcal{I}}\}$. It is easy to see that Γ_c is a $\mathcal{DL}$-type for $\mathfrak{C}$ and that $c \in C_{\Gamma_c}^{\mathcal{I}}$. This shows that $\bigcup_{\Gamma \in \mathfrak{T}(\mathfrak{C})} C_\Gamma^{\mathcal{I}} \equiv \Delta$ holds, and, thus, also $|\Delta| = \sum_{\Gamma \in \mathfrak{T}(\mathfrak{C})} |C_\Gamma^{\mathcal{I}}|$ because type concepts are pairwise disjoint. ☐

As a consequence of (7.12), we may say that an individual $c \in \Delta$ is of the $\mathcal{DL}$-type Γ in the $\mathcal{DL}$-interpretation $\mathcal{I} \in \mathfrak{I}_\Delta$ iff $c \in C_\Gamma^{\mathcal{I}}$. Further, $\mathcal{DL}$-types induce an equivalence relation on Δ so that two individuals are equivalent iff they are of the same $\mathcal{DL}$-type. With this, conditional structures $\sigma_{\mathcal{R}}(\mathcal{I})$ (cf. 7.10) can be broken down to the impacts of the $\mathcal{DL}$-types on $\sigma_{\mathcal{R}}(\mathcal{I})$.

Definition 7.2.4: Conditional Impact of $\mathcal{DL}$-Types

(cf. [Wilhelm et al., 2019a])

Let $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ be an $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base with $\mathcal{A} = \emptyset$, and let $\Gamma \in \mathfrak{T}_{\mathcal{R}}$. Then, we define the *conditional impact* of Γ for $\mathcal{R}$ by

$$\rho_{\mathcal{R}}(\Gamma) = \prod_{i=1}^{n} \begin{cases} \mathbf{a_i^+} & \text{if } \{C_i, D_i\} \subseteq \Gamma \\ \mathbf{a_i^-} & \text{if } \{C_i, \neg D_i\} \subseteq \Gamma \\ \mathbf{1} & \text{if } \{\neg C_i\} \subseteq \Gamma \end{cases}.$$

The conditional impacts of the $\mathcal{DL}$-types for $\mathcal{R}_W$ from Example 7.2.2 are shown in Table 7.2. Analogously to the conditional impact of a $\mathcal{DL}$-type, we may define the *features* of a $\mathcal{DL}$-type $\Gamma \in \mathfrak{T}_{\mathcal{R}}$ for $i = 1, \ldots, n$, i.e., for all $\mathcal{ALC}^{\mathsf{ME}}$-conditionals $(D_i|C_i)[p_i]$ from $\mathcal{R}$, by

$$f_i(\Gamma) = \begin{cases} 1 - p_i & \text{if } \{C_i, D_i\} \subseteq \Gamma \\ -p_i & \text{if } \{C_i, \neg D_i\} \subseteq \Gamma \\ 0 & \text{if } \{\neg C_i\} \subseteq \Gamma \end{cases}.$$

This definition, of course, refers to the feature functions defined in (7.5). Based on the notions of conditional impacts and features of $\mathcal{DL}$-types, we have the following characterization of conditional structures and feature function values of $\mathcal{DL}$-interpretations.

Proposition 7.2.5: Conditional Structures Based on $\mathcal{DL}$-Types
[Wilhelm et al., 2019a]

Let $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ be an $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base with $\mathcal{A} = \emptyset$. Then, for all $\mathcal{I} \in \mathfrak{I}_\Delta$,

- $\sigma_{\mathcal{R}}(\mathcal{I}) = \prod_{\Gamma \in \mathfrak{T}_{\mathcal{R}}} \rho_{\mathcal{R}}(\Gamma)^{|C_\Gamma^{\mathcal{I}}|}$,
- $f_i(\mathcal{I}) = \sum_{\Gamma \in \mathfrak{T}_{\mathcal{R}}} |C_\Gamma^{\mathcal{I}}| \cdot f_i(\Gamma)$ for $i = 1, \ldots, n$.

Proof. For $\mathcal{I} \in \mathfrak{I}_\Delta$, we have

$$\begin{aligned}
\sigma_{\mathcal{R}}(\mathcal{I}) &= \prod_{i=1}^{n} (\mathbf{a_i^+})^{|C_i^{\mathcal{I}} \cap D_i^{\mathcal{I}}|} \cdot (\mathbf{a_i^-})^{|C_i^{\mathcal{I}} \cap (\neg D_i)^{\mathcal{I}}|} \\
&= \prod_{i=1}^{n} (\mathbf{a_i^+})^{|(\bigsqcup_{\substack{\Gamma \in \mathfrak{T}_{\mathcal{R}}:\\ \{C_i, D_i\} \subseteq \Gamma}} C_\Gamma)^{\mathcal{I}}|} \cdot (\mathbf{a_i^-})^{|(\bigsqcup_{\substack{\Gamma \in \mathfrak{T}_{\mathcal{R}}:\\ \{C_i, \neg D_i\} \subseteq \Gamma}} C_\Gamma)^{\mathcal{I}}|} \\
&= \prod_{i=1}^{n} (\mathbf{a_i^+})^{\sum_{\substack{\Gamma \in \mathfrak{T}_{\mathcal{R}}:\\ C_i, D_i \subseteq \Gamma}} |C_\Gamma^{\mathcal{I}}|} \cdot (\mathbf{a_i^-})^{\sum_{\substack{\Gamma \in \mathfrak{T}_{\mathcal{R}}:\\ \{C_i, \neg D_i\} \subseteq \Gamma}} |C_\Gamma^{\mathcal{I}}|} \\
&= \prod_{i=1}^{n} \prod_{\Gamma \in \mathfrak{T}_{\mathcal{R}}} \begin{cases} (\mathbf{a_i^+})^{|C_\Gamma^{\mathcal{I}}|} & \text{if} \quad \{C_i, D_i\} \subseteq \Gamma \\ (\mathbf{a_i^-})^{|C_\Gamma^{\mathcal{I}}|} & \text{if} \quad \{C_i, \neg D_i\} \subseteq \Gamma \end{cases} \\
&= \prod_{\Gamma \in \mathfrak{T}_{\mathcal{R}}} \rho_{\mathcal{R}}(\Gamma)^{|C_\Gamma^{\mathcal{I}}|},
\end{aligned}$$

and also

$$\begin{aligned}
f_i(\mathcal{I}) &= |C_i^{\mathcal{I}} \cap D_i^{\mathcal{I}}| - p_i \cdot |C_i^{\mathcal{I}}| \\
&= \sum_{\substack{\Gamma \in \mathfrak{T}_{\mathcal{R}}:\\ \{C_i, D_i\} \subseteq \Gamma}} |C_\Gamma| - p_i \cdot \Big(\sum_{\substack{\Gamma \in \mathfrak{T}_{\mathcal{R}}:\\ \{C_i, D_i\} \subseteq \Gamma}} |C_\Gamma| + \sum_{\substack{\Gamma \in \mathfrak{T}_{\mathcal{R}}:\\ \{C_i, \neg D_i\} \subseteq \Gamma}} |C_\Gamma| \Big) \\
&= \sum_{\substack{\Gamma \in \mathfrak{T}_{\mathcal{R}}:\\ \{C_i, D_i\} \subseteq \Gamma}} |C_\Gamma| \cdot (1 - p_i) + \sum_{\substack{\Gamma \in \mathfrak{T}_{\mathcal{R}}:\\ \{C_i, \neg D_i\} \subseteq \Gamma}} |C_\Gamma| \cdot (-p_i) + \sum_{\substack{\Gamma \in \mathfrak{T}_{\mathcal{R}}:\\ \neg C_i \in \Gamma}} |C_\Gamma| \cdot 0 \\
&= \sum_{\Gamma \in \mathfrak{T}_{\mathcal{R}}} |C_\Gamma| \cdot f_i(\Gamma),
\end{aligned}$$

for $i = 1, \ldots, n$. □

Proposition 7.2.5 implies that $\mathcal{DL}$-interpretations $\mathcal{I} \in \mathfrak{I}_\Delta$ with the same counts $|C_\Gamma^{\mathcal{I}}|$ for $\Gamma \in \mathfrak{T}_{\mathcal{R}}$ constitute equivalence classes on $\mathfrak{I}_\Delta$. We define the equivalence relation $\sim_{\mathcal{R}}$ on $\mathfrak{I}_\Delta$ by $\mathcal{I} \sim_{\mathcal{R}} \mathcal{I}'$ iff $|C_\Gamma^{\mathcal{I}}| = |C_\Gamma^{\mathcal{I}'}|$ for all $\Gamma \in \mathfrak{T}_{\mathcal{R}}$ and obtain the following proposition.

Proposition 7.2.6: $\mathcal{DL}$-Type Equivalence and Maximum Entropy
[Wilhelm et al., 2019a]

Let $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ be an $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base with $\mathcal{A} = \emptyset$, and let $\mathcal{I}, \mathcal{I}' \in \mathfrak{I}_\Delta$ be $\mathcal{DL}$-interpretations with $\mathcal{I} \sim_{\mathcal{R}} \mathcal{I}'$. Then,

$$\sigma_{\mathcal{R}}(\mathcal{I}) = \sigma_{\mathcal{R}}(\mathcal{I}') \quad \text{and} \quad f_i(\mathcal{I}) = f_i(\mathcal{I}'), \quad i = 1, \ldots, n. \tag{7.13}$$

If, in addition, $\mathcal{I}, \mathcal{I}' \in \mathfrak{I}_\Delta^{\mathcal{R}}$ and $\mathcal{R}$ is p-consistent, then $\mathcal{P}^{\mathsf{ME}}_{\Delta,\mathcal{R}}(\mathcal{I}) = \mathcal{P}^{\mathsf{ME}}_{\Delta,\mathcal{R}}(\mathcal{I}')$ holds as well as $\mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}(\mathcal{I}) = \mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}(\mathcal{I}')$ for any approximation $\mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}$ to $\mathcal{P}^{\mathsf{ME}}_{\Delta,\mathcal{R}}$ defined by (7.8).

Proof. Let $\mathcal{I} \sim_{\mathcal{R}} \mathcal{I}'$. Then, by definition, $|C^{\mathcal{I}}_\Gamma| = |C^{\mathcal{I}'}_\Gamma|$ for all $\Gamma \in \mathfrak{T}_{\mathcal{R}}$ and, according to Proposition 7.2.5,

$$\sigma_{\mathcal{R}}(\mathcal{I}) = \prod_{\Gamma \in \mathfrak{T}_{\mathcal{R}}} \rho_{\mathcal{R}}(\Gamma)^{|C^{\mathcal{I}}_\Gamma|} = \prod_{\Gamma \in \mathfrak{T}_{\mathcal{R}}} \rho_{\mathcal{R}}(\Gamma)^{|C^{\mathcal{I}'}_\Gamma|} = \sigma_{\mathcal{R}}(\mathcal{I}').$$

Further, for $i = 1, \ldots, n$,

$$f_i(\mathcal{I}) = \sum_{\Gamma \in \mathfrak{T}_{\mathcal{R}}} |C^{\mathcal{I}}_\Gamma| \cdot f_i(\Gamma) = \sum_{\Gamma \in \mathfrak{T}_{\mathcal{R}}} |C^{\mathcal{I}'}_\Gamma| \cdot f_i(\Gamma) = f_i(\mathcal{I}').$$

Consequently,

$$\mathcal{P}^{\mathsf{ME}}_{\Delta,\mathcal{R}}(\mathcal{I}) = \alpha_0 \cdot \prod_{i=1}^{n} \alpha_i^{f_i(\mathcal{I})} = \alpha_0 \cdot \prod_{i=1}^{n} \alpha_i^{f_i(\mathcal{I}')} \mathcal{P}^{\mathsf{ME}}_{\Delta,\mathcal{R}}(\mathcal{I}'),$$

and

$$\mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}(\mathcal{I}) = \alpha^*_0 \cdot \prod_{i=1}^{n} (\alpha^*_i)^{f_i(\mathcal{I})} = \alpha^*_0 \cdot \prod_{i=1}^{n} (\alpha^*_i)^{f_i(\mathcal{I}')} \mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}(\mathcal{I}'). \qquad \square$$

We will use Proposition 7.2.6 to compute approximations of maximum entropy probabilities on the level of the equivalence classes of $\mathcal{DL}$-interpretations induced by $\sim_{\mathcal{R}}$. Because the number of these equivalence classes $[\mathcal{I}]_{\sim_{\mathcal{R}}}$ is bounded by $(k+1)^{|\mathfrak{T}_{\mathcal{R}}|}$ which is polynomial in $k = |\Delta|$ this is an important step towards lifted inference in $\mathcal{ALC}^{\mathsf{ME}}$. The upper bound of the number of equivalence classes holds because the equivalence classes $[\mathcal{I}]_{\sim_{\mathcal{R}}}$ can differ in the numbers $|C^{\mathcal{I}}_\Gamma|$ for $\Gamma \in \mathfrak{T}_{\mathcal{R}}$ all of which can vary between zero and k. Next, we give combinatorial arguments that allow us to compute the equivalence classes and their cardinalities in time polynomial in k.

▶ Counting Strategies for Lifted Inference in $\mathcal{ALC}^{\mathsf{ME}}$

For our strategy on drawing lifted inferences from $\mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}$ it is essential to determine the equivalence classes with respect to the equivalence relation $\sim_{\mathcal{R}}$ on $\mathfrak{I}_\Delta$ as well as their cardinalities efficiently, where $\sim_{\mathcal{R}}$ is defined by $\mathcal{I} \sim_{\mathcal{R}} \mathcal{I}'$ iff $|C^{\mathcal{I}}_{\Gamma}| = |C^{\mathcal{I}'}_{\Gamma}|$ for $\Gamma \in \mathfrak{T}_{\mathcal{R}}$. First, we show how to set up the equivalence classes and then develop a calculation rule for their cardinalities.

The equivalence classes $[\mathcal{I}]_{\sim_{\mathcal{R}}}$ differ in the number of individuals from Δ which belong to $C^{\mathcal{I}}_{\Gamma}$ for $\Gamma \in \mathfrak{T}_{\mathcal{R}}$. Thus, specifying the equivalence classes is related to the combinatorial problem of classifying k-many elements into $|\mathfrak{T}_{\mathcal{R}}|$-many categories ($k = |\Delta|$). Let $\mathfrak{T}_{\mathcal{R}} = \{\Gamma_1, \ldots, \Gamma_m\}$ with $k_i = |C^{\mathcal{I}}_{\Gamma_i}|$ for $i = 1, \ldots, m$ for a fixed $\mathcal{DL}$-interpretation $\mathcal{I}$. Then, $[\mathcal{I}]_{\sim_{\mathcal{R}}}$ is in a one-to-one correspondence with the vector $\vec{k} = (k_1, \ldots, k_m) \in \mathbb{N}_0^m$, and we may define $[\mathcal{I}]_{\vec{k}}$ as the unique equivalence class corresponding to $\vec{k}$. Because the $\mathcal{DL}$-types partition Δ (cf. (7.12)), for all equivalence classes $[\mathcal{I}]_{\vec{k}}$ we have

$$\sum_{i=1}^{m} k_i = k. \tag{7.14}$$

However, not every vector $\vec{k} \in \mathbb{N}_0^m$ which satisfies (7.14) leads to an equivalence class $[\mathcal{I}]_{\sim_{\mathcal{R}}}$. This is because existential restrictions relate individuals to each other and may force the existence of further individuals of a certain $\mathcal{DL}$-type. We give an example.

Example 7.2.7: *[Wilhelm et al., 2019a]*

We consider the $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base

$$\mathcal{R}_{smk} = (\emptyset, \emptyset, \{(\exists \mathsf{friend}.\mathsf{Smoker} | \mathsf{Smoker})[0.9]\},$$

stating that smokers typically have at least one friend who is also a smoker. There are the following four $\mathcal{DL}$-types for $\mathcal{R}_{smk}$ in which we abbreviate concept and role names by their first letter:

$$\Gamma_1 = \{S, \exists f.S\}, \quad \Gamma_2 = \{S, \neg\exists f.S\}, \quad \Gamma_3 = \{\neg S, \exists f.S\}, \quad \Gamma_4 = \{\neg S, \neg\exists f.S\}.$$

If there is an individual of type Γ_3, i.e., a non-smoker who has a friend that smokes, then there must be a second person who is a smoker, i.e., an individual of type Γ_1 or Γ_2. Thus, $k_3 > 0$ enforces $k_1 + k_2 > 0$.

We can deal with this phenomenon by exploiting the following definition from [Baader and Ecke, 2017].

Definition 7.2.8: Satisfaction of Existential Restrictions

[Baader and Ecke, 2017]

Let Γ be a $\mathcal{DL}$-type which contains an existential restriction $\exists r.A$, and let $\neg\exists r.B_1, \ldots, \neg\exists r.B_l$ be all the negated existential restrictions for the same role name r in Γ. Then, a $\mathcal{DL}$-type Γ' *satisfies* $\exists r.A$ iff $\{A, \neg B_1, \ldots, \neg B_l\} \subseteq \Gamma'$.

For every $\mathcal{DL}$-type $\Gamma \in \mathfrak{T}_{\mathcal{R}}$ and for every existential restriction $\exists r.A \in \Gamma$, we have

$$k(\Gamma) = 0 \quad \text{or} \quad \sum_{\substack{\Gamma' \in \mathfrak{T}_{\mathcal{R}}: \\ \Gamma' \text{ satisfies } \exists r.A \text{ in } \Gamma}} k(\Gamma') > 0, \tag{7.15}$$

which generalizes our observation in Example 7.2.7. In addition, in Baader and Ecke [2017] it is shown that any vector $\vec{k}$ which satisfies (7.14) and (7.15) is realized by a $\mathcal{DL}$-interpretation. Thus, the set of all equivalence classes induced by $\sim_{\mathcal{R}}$ is

$$\mathfrak{I}_{\Delta}/{\sim_{\mathcal{R}}} = \{[\mathcal{I}]_{\vec{k}} \mid \vec{k} \in \mathbb{N}_0^m \colon (7.14) \text{ and } (7.15) \text{ hold}\}. \tag{7.16}$$

The characterization (7.16) of the equivalence classes induced by $\sim_{\mathcal{R}}$ allows us to enumerate them in time polynomial in $k = |\Delta|$ because (7.15) is independent of Δ and iterating through all vectors $\vec{k} \in \mathbb{N}_0^m$ which satisfy (7.14) is possible in time polynomial in k since m is constant in k.

Note that we are actually interested only in those $\mathcal{DL}$-interpretations that satisfy all general concept inclusions in $\mathcal{T}$. In these $\mathcal{DL}$-interpretations there must not exist an individual $c \in \Delta$ with $c \in C^{\mathcal{I}}$ and $c \notin D^{\mathcal{I}}$ for any $C \sqsubseteq D \in \mathcal{T}$. Due to Proposition 7.2.3, this constraint is equivalent to

$$\forall C \sqsubseteq D \in \mathcal{T} \quad \forall \Gamma \in \mathfrak{T}_{\mathcal{R}}\colon \quad \{C, \neg D\} \subseteq \Gamma \quad \text{implies} \quad k(\Gamma) = 0. \tag{7.17}$$

We say that a $\mathcal{DL}$-type $\Gamma \in \mathfrak{T}_{\mathcal{R}}$ for which $C \in \Gamma$ implies $D \in \Gamma$ for all $C \sqsubseteq D \in \mathcal{T}$ *satisfies* $\mathcal{T}$, written $\Gamma \models \mathcal{T}$. Hence, (7.17) enforces that $k(\Gamma) > 0$ holds for only those $\mathcal{DL}$-types which satisfy $\mathcal{T}$. Consequently, the set of all equivalence classes of those $\mathcal{DL}$-interpretations that satisfy $\mathcal{T}$, i.e., which are in $\mathfrak{I}_{\Delta}^{\mathcal{R}}$, is

$$\mathfrak{I}_{\Delta}^{\mathcal{R}}/{\sim_{\mathcal{R}}} = \{[\mathcal{I}]_{\vec{k}} \in \mathfrak{I}_{\Delta}/{\sim_{\mathcal{R}}} \mid (7.17) \text{ holds}\}.$$

This set can be determined in time polynomial in $|\Delta|$, too, because (7.17) is independent of Δ.

It remains to determine the cardinalities $|[\mathcal{I}]_{\vec{k}}|$ of the equivalence classes $[\mathcal{I}]_{\vec{k}} \in \mathfrak{I}_{\Delta}^{\mathcal{R}}/{\sim_{\mathcal{R}}}$. These cardinalities depend on two factors. First, the k individuals in Δ have to be allocated to the $\mathcal{DL}$-types in $\mathfrak{T}_{\mathcal{R}}$. This is the combinatorial problem of classifying elements into categories mentioned at the beginning of this section, and for which there are

$$\binom{k}{k_1, \dots, k_m} = \frac{k!}{k_1! \cdots k_m!}$$

many possibilities if $k_i = |\Gamma_i|$ for $\Gamma_i \in \mathfrak{T}_{\mathcal{R}}$.

Second, recall that a $\mathcal{DL}$-interpretation $\mathcal{I} \in \mathfrak{I}_\Delta$ is fully specified iff for all concept names $C \in \mathcal{N}_C$ and for all role names $r \in \mathcal{N}_R$ the sets $C^{\mathcal{I}}$ and $r^{\mathcal{I}}$ are fixed (because individual names are mapped to themselves in $\mathcal{ALC}^{\mathsf{ME}}$). As every concept name A that is mentioned in $\mathcal{R}$ also occurs in every single type for $\mathcal{R}$, either positive or negated, the sets $A^{\mathcal{I}}$ for these concept names are uniquely determined by the types. However, this does not hold for concept names that are not mentioned in $\mathcal{R}$. Given a concept name in $\mathcal{N}_C \backslash \mathsf{sig}_C(\mathcal{R})$, one can choose freely for every individual in Δ whether it belongs to this concept name or not. In total, there are $2^{k \cdot |\mathcal{N}_C \backslash \mathsf{sig}_C(\mathcal{R})|}$ possibilities of allocating the k individuals in Δ to the concept names in $\mathcal{N}_C \setminus \mathsf{sig}_C(\mathcal{R})$.

Determining the degree of freedom that arises from role memberships that are not specified is more difficult. Again, for the role names that are not mentioned in $\mathcal{R}$, there is free choice such that there are $2^{k^2 \cdot |\mathcal{N}_R \backslash \mathsf{sig}_R(\mathcal{R})|}$ possible combinations of allocating k^2 many tuples of individuals to them. Computing the possibilities to allocate tuples of individuals to role names that are mentioned in $\mathcal{R}$ is a bit more difficult, though. For this purpose, we formally define the *degree of freedom* of a role name.

Definition 7.2.9: Degree of Freedom of Role Names

[Wilhelm et al., 2019a]

Let $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ be an $\mathcal{ALC}^{\mathsf{ME}}$-knowledge base with $\mathcal{A} = \emptyset$, and let $\mathfrak{T}_{\mathcal{R}} = \{\Gamma_1, \dots, \Gamma_m\}$ be the set of $\mathcal{DL}$-types for $\mathcal{R}$ with $k_i = |C^{\mathcal{I}}_{\Gamma_i}|$ for $i = 1, \dots, m$ for a fixed $\mathcal{DL}$-interpretation $\mathcal{I}$. Further let $\Gamma_i \in \mathfrak{T}_{\mathcal{R}}$ a $\mathcal{DL}$-type for $\mathcal{R}$, and let $\exists r.A_1, \dots, \exists r.A_l$ be all the existential restrictions and let $\neg\exists r.B_1, \dots, \neg\exists r.B_h$ be all the negated existential restrictions for the role name r in Γ_i. We define the *degree of freedom* of r in Γ_i with respect to $[\mathcal{I}]_{\vec{k}} \in \mathfrak{I}_\Delta / {\sim_{\mathcal{R}}}$ as

$$\phi_{\vec{k}}(\Gamma_i, r) = \Big(\sum_{\mathfrak{J} \subseteq \{1,\dots,l\}} (-1)^{|\mathfrak{J}|} \cdot \prod_{\substack{j=1,\dots,m: \\ \{\neg B_1,\dots,\neg B_h\} \subseteq \Gamma_j, \\ \forall i \in \mathfrak{J}:\, \neg A_i \in \Gamma_j}} 2^{k_j} \Big)^{k_i}. \tag{7.18}$$

Definition 7.2.9 is a generalization of Definition 7.2.8 that takes counting aspects into account by making use of the well-known inclusion-exclusion principle from combinatorics (cf. [Roberts and Tesman, 2009] and Definition 7.2.9 in the appendix). In this way, it keeps track of which individual guarantees that a certain existential restriction holds. To understand Definition 7.2.9, assume that there is *no* positive existential restriction $\exists r.A$ for r in Γ_i. Then, for every individual $c \in C^{\mathcal{I}}_{\Gamma_i}$ and for every individual c' in any $C^{\mathcal{I}}_{\Gamma_j}$ with $\{\neg B_1, \dots, \neg B_h\} \subseteq \Gamma_j$, whether $(c, c') \in r^{\mathcal{I}}$ or not can be chosen freely, which results in the factor $(2^{k_j})^{k_i}$ in $\phi_{\vec{k}}(\Gamma_i, r)$. Now, assume there is one positive existential restriction $\exists r.A$ in Γ_i. For individuals $c' \in C^{\mathcal{I}}_{\Gamma_j}$ with Γ_j such that $\{\neg A, \neg B_1, \dots, \neg B_h\} \subseteq \Gamma_j$, again the belonging of (c, c') to $r^{\mathcal{I}}$ is unrestricted. However, there must be at least one individual c'' among the individuals of a $\mathcal{DL}$-type Γ_h with $\{A, \neg B_1, \dots, \neg B_h\} \subseteq \Gamma_h$ such that $(c, c'') \in r^{\mathcal{I}}$. This results in the degree of freedom

$$\begin{aligned}
\phi_{\vec{k}}(\Gamma_i, r) &= \Big(\prod_{\substack{\Gamma_j \in \mathfrak{T}_{\mathcal{R}}:\\ \{\neg B_1, \dots, \neg B_h\} \subseteq \Gamma_j}} 2^{k_j} - \prod_{\substack{\Gamma_j \in \mathfrak{T}_{\mathcal{R}}:\\ \{\neg B_1, \dots, \neg B_h, \neg A\} \subseteq \Gamma_j}} 2^{k_j} \Big)^{k_i} \\
&= \Big((\prod_{\substack{\Gamma_j \in \mathfrak{T}_{\mathcal{R}}:\\ \{\neg B_1, \dots, \neg B_h, \neg A\} \subseteq \Gamma_j}} 2^{k_j}) \cdot (\prod_{\substack{\Gamma_j \in \mathfrak{T}_{\mathcal{R}}:\\ \{\neg B_1, \dots, \neg B_h, A\} \subseteq \Gamma_j}} 2^{k_j} - 1) \Big)^{k_i}.
\end{aligned}$$

If there are multiple positive existential restrictions, then all of them could be satisfied by the same tuple of individuals. Alternatively, there may exist several tuples of individuals each satisfying only some of the restrictions. Then, a combination of tuples is needed to satisfy all of the existential restrictions. This makes the application of the inclusion-exclusion principle necessary. Altogether, for every $[\mathcal{I}]_{\vec{k}} \in \mathfrak{I}_\Delta / {\sim_{\mathcal{R}}}$, one has

$$|[\mathcal{I}]_{\vec{k}}| = \binom{k}{k_1, \dots, k_m} \cdot \Big(\prod_{j=1}^{m} \prod_{r \in \mathsf{sig}_R(\mathcal{R})} \phi_{\vec{k}}(\Gamma_j, r) \Big) \cdot 2^{(|\mathcal{N}_C \setminus \mathsf{sig}_C(\mathcal{R})|) \cdot k} \cdot 2^{(|\mathcal{N}_R \setminus \mathsf{sig}_R(\mathcal{R})|) \cdot k^2},$$

which can be calculated in time polynomial in $k = |\Delta|$.

From these deliberations it follows that drawing inferences in $\mathcal{ALC}^{\mathsf{ME}}$ from $\mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}$ is domain-liftable, which we state in the next proposition. In the proof of this proposition we exploit that the probabilities of the conditionals in $\mathcal{R}$ are rational numbers.

Proposition 7.2.10: Domain-Lifted Inferences in $\mathcal{ALC}^{\mathsf{ME}}$

[Wilhelm et al., 2019a]

Let $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ be a consistent knowledge base with $\mathcal{A} = \emptyset$, and let $\mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}$ be a probability distribution with $\vec{\alpha}^* \in \mathbb{Q}^n_{>0}$ defined as in (7.8). Further, let C and D be $\mathcal{ALC}^{\mathsf{ME}}$-concepts. Then, calculating the probability p for which $\mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}} \models (D|C)[p]$ holds, is possible in time polynomial in $k = |\Delta|$.

Proof (from [Wilhelm et al., 2019a]). For the probabilities of the conditionals in $\mathcal{C}$, we have $p_i = \frac{s_i}{t_i}$ with $s_i, t_i \in \mathbb{N}_{>0}$ for $i = 1, \dots, n$. Further, we write $q = (D|C)[p]$.

Then,

$$\begin{aligned}
p &= \frac{\sum_{\mathcal{I}\in\mathfrak{I}_\Delta} |C^{\mathcal{I}} \cap D^{\mathcal{I}}| \cdot \mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}(\mathcal{I})}{\sum_{\mathcal{I}\in\mathfrak{I}_\Delta} |C^{\mathcal{I}}| \cdot \mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}(\mathcal{I})} \\
&= \frac{\sum_{\mathcal{I}\in\mathfrak{I}^{\mathcal{R}}_\Delta} |C^{\mathcal{I}} \cap D^{\mathcal{I}}| \cdot \alpha^*_0 \cdot \prod_{i=1}^n (\alpha^*_i)^{f_i(\mathcal{I})}}{\sum_{\mathcal{I}\in\mathfrak{I}^{\mathcal{R}}_\Delta} |C^{\mathcal{I}}| \cdot \alpha^*_0 \cdot \prod_{i=1}^n (\alpha^*_i)^{f_i(\mathcal{I})}} \\
&= \frac{\sum_{\mathcal{I}\in\mathfrak{I}^{\mathcal{R}}_\Delta} |C^{\mathcal{I}} \cap D^{\mathcal{I}}| \cdot \prod_{i=1}^n (\alpha^*_i)^{|C^{\mathcal{I}}\cap D^{\mathcal{I}}| - \frac{s_i}{t_i}\cdot|C^{\mathcal{I}}|}}{\sum_{\mathcal{I}\in\mathfrak{I}^{\mathcal{R}}_\Delta} |C^{\mathcal{I}}| \cdot \prod_{i=1}^n (\alpha^*_i)^{|C^{\mathcal{I}}\cap D^{\mathcal{I}}| - \frac{s_i}{t_i}\cdot|C^{\mathcal{I}}|}} \\
&= \frac{\sum_{\mathcal{I}\in\mathfrak{I}^{\mathcal{R}}_\Delta} |C^{\mathcal{I}} \cap D^{\mathcal{I}}| \cdot \prod_{i=1}^n (\alpha^*_i)^{t_i\cdot|C^{\mathcal{I}}\cap D^{\mathcal{I}}| - s_i\cdot|C^{\mathcal{I}}| + s_i\cdot|\Delta|}}{\sum_{\mathcal{I}\in\mathfrak{I}^{\mathcal{R}}_\Delta} |C^{\mathcal{I}}| \cdot \prod_{i=1}^n (\alpha^*_i)^{t_i\cdot|C^{\mathcal{I}}\cap D^{\mathcal{I}}| - s_i\cdot|C^{\mathcal{I}}| + s_i\cdot|\Delta|}}.
\end{aligned} \tag{7.19}$$

Note that

$$t_i \cdot |C_i^{\mathcal{I}} \cap D_i^{\mathcal{I}}| - s_i \cdot |C_i^{\mathcal{I}}| + s_i \cdot |\Delta| \geq 0$$

for all $\mathcal{I} \in \mathfrak{I}^{\mathcal{R}}_\Delta$ and $i = 1, \ldots, n$. Hence, the last fraction in (7.19) mentions sums over products of integers ($|C^{\mathcal{I}} \cap D^{\mathcal{I}}|$ and $|C^{\mathcal{I}}|$, respectively) and rational numbers (α^*_i) with integer exponents only.

It remains to show that (7.19) can be calculated in time polynomial in $|\Delta|$. To prove this, we aggregate the $\mathcal{DL}$-interpretations into equivalence classes. However, we have to modify the set of $\mathcal{DL}$-types the equivalence classes are based on since the query conditional q may mention additional concepts that are not considered by the $\mathcal{DL}$-types in $\mathfrak{T}_{\mathcal{R}}$. Let $\mathfrak{C}_q$ be the closure of $\mathfrak{C}^+_q = \{E \mid E \in \mathsf{sub}(C) \cup \mathsf{sub}(D)\}$ under negation, and let $\mathfrak{T}^q_{\mathcal{R}} = \mathfrak{T}(\mathfrak{C}_{\mathcal{R}} \cup \mathfrak{C}_q)$. For $\mathcal{DL}$-interpretations $\mathcal{I}, \mathcal{I}' \in \mathfrak{I}_\Delta$, we define the equivalence relation $\mathcal{I} \sim^q_{\mathcal{R}} \mathcal{I}'$ iff $C^{\mathcal{I}}_\Gamma = C^{\mathcal{I}'}_\Gamma$ for all $\Gamma \in \mathfrak{T}^q_{\mathcal{R}}$ in analogy to $\sim_{\mathcal{R}}$. Every $\mathcal{DL}$-type $\Gamma \in \mathfrak{T}^q_{\mathcal{R}}$ is a refinement of a unique $\mathcal{DL}$-type $\Gamma' \in \mathfrak{T}_{\mathcal{R}}$, i.e. $\Gamma' \subseteq \Gamma$, and we may define $\rho_{\mathcal{R}}(\Gamma') = \rho_{\mathcal{R}}(\Gamma)$. In plain words, Γ' inherits its conditional impact from Γ. Accordingly, we define $f_i(\Gamma') = f_i(\Gamma)$ for $i = 1, \ldots, n$. Then Proposition 7.2.5 and Proposition 7.2.6 still hold when replacing the underlying set of types $\mathfrak{T}_{\mathcal{R}}$ by $\mathfrak{T}^q_{\mathcal{R}}$. Also, the counting strategies and the complexity results for $\mathfrak{I}_\Delta/\sim^q_{\mathcal{R}}$ are the same as for $\mathfrak{I}_\Delta/\sim_{\mathcal{R}}$, and (7.19) can be rewritten to

$$p = \frac{\sum_{[\mathcal{I}]_{\vec{k}} \in \mathfrak{I}^{\mathcal{R}}_\Delta/\sim^q_{\mathcal{R}}} k^+_i \cdot \prod_{j=1}^n (\alpha^*_j)^{t_j\cdot k^+_j - s_j\cdot k^o_j + s_j\cdot|\Delta|}}{\sum_{[\mathcal{I}]_{\vec{k}} \in \mathfrak{I}^{\mathcal{R}}_\Delta/\sim^q_{\mathcal{R}}} k^o_i \cdot \prod_{j=1}^n (\alpha^*_j)^{t_j\cdot k^+_j - s_j\cdot k^o_j + s_j\cdot|\Delta|}}$$

where $k^+_i = \sum_{\Gamma_j \in \mathfrak{T}^q_{\mathcal{R}}\colon \{C_i, D_i\} \subseteq \Gamma_j} k_j$ and $k^o_i = \sum_{\Gamma_j \in \mathfrak{T}^q_{\mathcal{R}}\colon C_i \in \Gamma_j} k_j$. This fraction can be calculated in time polynomial in $k = |\Delta|$. □

Proposition 7.2.10 states that inferences in $\mathcal{ALC}^{\mathsf{ME}}$ from $\mathcal{P}_{\vec{\alpha}^*_{\Delta,\mathcal{R}}}$ are domain-lifted (cf. Definition 5.1.1). Hence, the message of Proposition 7.2.10 is that the crucial part of drawing inferences at maximum entropy from $\mathcal{R}$ according to (7.2) is the approximation of $\mathcal{P}^{\mathsf{ME}}$ which brings us back to condensed iterative scaling (cf. Section 5.2). Once this approximation is given, all further calculations can be performed in time polynomial in $k = |\Delta|$. Note that the consistency of $\mathcal{R}$ can also be checked in time polynomial in k. We refer to [Wilhelm et al., 2019a] for the details.

7.3 $\mathcal{ALC}^{\mathsf{ME}}$ With Linear Arithmetic Constraints

In this section, we extend the Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ by *linear arithmetic constraints* to the Description Logic $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$ in order to be able to formulate beliefs over infinite domains. While Description Logics are usually defined with respect to domains which may be both finite or (countably) infinite, the Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ is restricted to a fixed finite domain. In the following, we discuss why a direct extension of $\mathcal{ALC}^{\mathsf{ME}}$ to countably infinite domains is challenging and propose with the incorporation of linear arithmetic constraints a notable alternative. The advantage of our approach is that the satisfiability of the linear arithmetic constraints is checked modulo theory such that the probability space remains finite. In a propositional setting this approach was published in [Wilhelm, 2023] first and is adapted to the description logical context here.

▶ Maximum Entropy and Countably Infinite Domains

A popular way of approaching countably infinite domains in relational maximum entropy reasoning is to treat the (finite) domain size $k = |\mathsf{Const}_\Sigma|$ as a parameter and consider limit processes for k tending towards infinity [Barnett and Paris, 2008; Thimm and Kern-Isberner, 2012]. This approach is "motivated by the idea that a countably infinite universe might be an idealization from a finite but inestimably large universe and that the rational beliefs assigned to the infinite universe should therefore be the limits of their finite counterparts" [Barnett and Paris, 2008]. We consider such a limit process in the most simple case, namely in the absence of any prior beliefs.

Example 7.3.1

Let $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ be a (relational) finite signature. The absence of beliefs here means that we consider the knowledge base $\mathcal{R} = (\emptyset, \emptyset)$. Recall that the maximum entropy Σ-model of $\mathcal{R}$ is the uniform distribution on the possible worlds in $\Omega(\Sigma)$ (cf. Proposition 4.2.2), i.e.,

$$\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) = \frac{1}{|\Omega(\Sigma)|}, \qquad \omega \in \Omega(\Sigma).$$

Hence, if the domain size $k = |\mathsf{Const}_\Sigma|$ tends towards infinity (and Pred_Σ does not consist of nullary predicates only), then the maximum entropy probabilities tend towards 0. As this holds for all possible worlds, one does not obtain a probability distribution in the limit because the probabilities do not sum up to 1. The underlying problem here is that there is no uniform distribution on countably infinite sample spaces.

Example 7.3.1 shows that the nonexistence of the uniform distribution on countably infinite sample spaces causes that a direct application of the principle of maximum entropy to countably infinite domains without any additional assumptions (like assumptions of independence, cf. [Barnett and Paris, 2008]) is not satisfactory in general. We give a further example which shows that this is not just an artifact but also affects reasoning with prior beliefs.

Example 7.3.2
Let $\Sigma = (\mathsf{Const}_\Sigma, \mathsf{Pred}_\Sigma)$ be a finite signature with $\mathsf{Const}_\Sigma = \{c_1, \dots, c_k\}$ and $\mathsf{Pred}_\Sigma = \{A/1\}$. We consider the knowledge base $\mathcal{R} = (\emptyset, \{r\})$ with

$$r = (\forall X.A(X)|\top)[p]$$

where p is an arbitrary probability. In the possible world $\omega' = \bigwedge_{i=1}^{k} A(c_i)$ the conditional r is verified. In all remaining possible worlds $\omega \in \Omega(\Sigma) \setminus \{\omega'\}$ the conditional r is falsified. Hence, the conditional structures of the possible worlds in $\Omega(\Sigma) \setminus \{\omega'\}$ are the same, and, because the maximum entropy model satisfies the property of conditional indifference (Proposition 4.3.5), $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega_1) = \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega_2)$ for $\omega_1, \omega_2 \in \Omega(\Sigma) \setminus \{\omega'\}$ follows. As a consequence,

$$\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega) = \frac{1 - \mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega')}{|\Omega(\Sigma)| - 1}, \qquad \omega \in \Omega(\Sigma) \setminus \{\omega'\}.$$

When k is tending towards infinity, we obtain that the MaxEnt probabilities of the worlds which falsify r tend to 0. As a consequence, $\mathcal{P}^{\mathsf{ME}}_{\Sigma,\mathcal{R}}(\omega')$ must tend to 1 in order to obtain a probability distribution in the limit. This, however, implies that $\mathcal{R}$ has a maximum entropy model in the limit only if $p = 1$.

An alternative way of approaching maximum entropy models for countably infinite domains is to reformulate the notion of maximum entropy. In [Williamson, 2008; Landes, 2021, 2023], comparative notions of *maximal entropy* are proposed. A drawback of these approaches is that the maximal entropy model, in general, is not unique, and there is no constructive method for computing maximal entropy models yet. Therefore, in the following, we suggest a further approach for linking countably infinite domains and maximum entropy reasoning which is based on linear arithmetic, as a background theory, and the concept of satisfiability modulo theory.

▶ Logic of Linear Arithmetic Constraints

As preparatory work for integrating linear arithmetic constraints into $\mathcal{ALC}^{\mathsf{ME}}$, we consider a *propositional language of linear arithmetic constraints* first. For this, we follow the elaborations in [Wilhelm, 2023]. With the term *linear arithmetic constraint*, we subsume all mathematical equations and inequations which are built of linear combinations over the integer numbers or real numbers.

Definition 7.3.3: Linear Arithmetic Constraint

(cf. [Barrett et al., 2021])

With *linear arithmetic constraint*, or *constraint* for short, we refer to a mathematical expression of the form

$$a_1 \cdot x_1 + \ldots + a_m \cdot x_m \bowtie a_0,$$

where $m \in \mathbb{N}_{>0}$, the x_i's are numerical variables, the a_i's are constants from $\mathbb{Z}$ or $\mathbb{R}$, and where $\bowtie \in \{<, \leq, =, \neq, \geq, >\}$.

Depending on whether the variables and constants range over the integers or reals, we name the arithmetic of such constraints either *linear integer arithmetic* and abbreviate it with $\mathcal{LIA}$, or we call it *linear real arithmetic*, $\mathcal{LRA}$ for short. Note that albeit both arithmetics are defined over infinite domains, the countably infinite set of integers $\mathbb{Z}$ or the uncountably infinite set of reals $\mathbb{R}$, they are decidable. More precisely, the satisfiability of any finite set of constraints from $\mathcal{LRA}$ can be decided in polynomial time [Karmarkar, 1984] whereas the satisfiability of $\mathcal{LIA}$-constraints is NP-complete [Papadimitriou, 1981]. The arithmetic $\mathcal{LIA}$ is also known as *Presburger arithmetic* [Presburger, 1929].

With $\mathsf{Sol}(\mathcal{C})$ we denote the set of the *common solutions* of a set of constraints $\mathcal{C}$.

Example 7.3.4

We consider the arithmetic constraints

$$\begin{aligned} p&: & x - y &> 1,000,000, \\ q&: & y &\geq 0, \\ r&: & x &= 10,000,000, \end{aligned}$$

in $\mathcal{LIA}$. The set of the common solutions is

$$\mathsf{Sol}(\{p, q, r\}) = \{(x, y) \in \mathbb{Z}^2 \mid x = 10,000,000, \quad y \in [0; 9,000,000)\}.$$

In our extension of $\mathcal{ALC}^{\mathsf{ME}}$, we want to allow for linear arithmetic constraints from both $\mathcal{LIA}$ and $\mathcal{LRA}$. For this, let $\mathcal{LA} = \mathcal{LIA} \cup \mathcal{LRA}$ be the set of constraints from either $\mathcal{LIA}$ or $\mathcal{LRA}$. For each constraint $c \in \mathcal{LA}$ we introduce a fresh symbol a_c and subsume all these symbols within the (uncountably infinite) set $\Sigma_{\mathcal{LA}}$. We call a_c the *constraint representative* of c.

Definition 7.3.5: Propositional Language of Constraints

[Wilhelm, 2023]

Let $\Sigma^f_{\mathcal{LA}} \subseteq \Sigma_{\mathcal{LA}}$ be a finite set of constraint representatives. We call the propositional language $\mathcal{L}(\Sigma^f_{\mathcal{LA}})$ in which the constraint representatives from $\Sigma^f_{\mathcal{LA}}$ play the role of the propositional atoms the *propositional language of constraints* over $\Sigma^f_{\mathcal{LA}}$.

Formulas in $\mathcal{L}(\Sigma^f_{\mathcal{LA}})$ and their interpretations are defined as usual in propositional logic. However, if $\mathcal{L}(\Sigma^f_{\mathcal{LA}})$ shall be coherent with $\mathcal{LA}$, we have to have a closer look at the semantics of formulas in $\mathcal{L}(\Sigma^f_{\mathcal{LA}})$ because the constraint representatives in $\Sigma^f_{\mathcal{LA}}$ are not independent when understood as constraints in $\mathcal{LA}$ in general. Instead, the satisfiability of these constraints can depend on each other. For instance, for each constraint

$$a_1 \cdot x_1 + \ldots + a_m \cdot x_m \bowtie a_0$$

in $\mathcal{LA}$, there is a *complementary constraint*

$$a_1 \cdot x_1 + \ldots + a_m \cdot x_m \ \phi(\bowtie)\ a_0$$

where ϕ maps $<$ to $\geq$, $\leq$ to $>$, $=$ to $\neq$, and vice versa. We denote the *complement* of a constraint $c \in \mathcal{LA}$ with $\widehat{c}$. Obviously, the solution sets of a constraint c and its complement $\widehat{c}$ are disjoint, i.e., $\mathsf{Sol}(\{c, \widehat{c}\}) = \emptyset$. Contrary to that, if $a_c, a_{\widehat{c}} \in \Sigma^f_{\mathcal{LA}}$, then there is an interpretation $I \in \mathsf{Int}(\Sigma^f_{\mathcal{LA}})$ with $I(a_c) = 1$ and $I(a_{\widehat{c}}) = 1$ which states that a_c and $a_{\widehat{c}}$ hold at the same time. In order to prevent such unwanted interpretations, we introduce a notion of coherence between $\mathcal{L}(\Sigma^f_{\mathcal{LA}})$ and $\mathcal{LA}$ on the level of interpretations.

Definition 7.3.6: Coherent Interpretation *[Wilhelm, 2023]*

Let $\Sigma^f_{\mathcal{LA}}$ be a finite set of constraint representatives, and, for $I \in \mathsf{Int}(\Sigma^f_{\mathcal{LA}})$, let

$$\mathcal{LA}(I) = \{c \mid a_c \in \Sigma^f_{\mathcal{LA}} \colon I(a_c) = 1\} \cup \{\widehat{c} \mid a_c \in \Sigma^f_{\mathcal{LA}} \colon I(a_c) = 0\} \tag{7.20}$$

be the translation of the interpretation I to the corresponding set of constraints from $\mathcal{LA}$. We call I *coherent* iff $\mathcal{LA}(I)$ is satisfiable. With $\mathsf{Int}_c(\Sigma^f_{\mathcal{LA}})$ we denote the set of all coherent interpretations over $\Sigma^f_{\mathcal{LA}}$.

Coherent interpretations respect $\mathcal{LA}$ in that they are not self-contradictory when understood as constraint sets. That is, $\mathsf{Sol}(\mathcal{LA}(I)) \neq \emptyset$ holds iff $I \in \mathsf{Int}_c(\Sigma^f_{\mathcal{LA}})$. In particular, there is no coherent interpretation $I \in \mathsf{Int}_c(\Sigma^f_{\mathcal{LA}})$ with $I(a_c) = I(a_{\widehat{c}}) = 1$ (or with $I(a_c) = I(a_{\widehat{c}}) = 0$) because $\mathsf{Sol}(\{c, \widehat{c}\}) = \emptyset$. As a consequence, $\neg a_c \equiv_C a_{\widehat{c}}$ holds where $\equiv_C$ means the logical equivalence defined wrt. the coherent interpretations only.

Definition 7.3.7: Logical Equivalence Modulo Coherence

Let $\Sigma^f_{\mathcal{LA}}$ be a finite set of constraint representatives, and let $a_c, a_d \in \Sigma^f_{\mathcal{LA}}$. Then, we say that a_c and a_d are *logically equivalent modulo coherence*, in symbols $a_c \equiv_C a_d$, iff $I(a_c) = I(a_d)$ holds for all coherent interpretations $I \in \mathsf{Int}_c(\Sigma^f_{\mathcal{LA}})$.

We now have a closer look at what the restriction to coherent interpretations implies. For this, we establish the notion of *admissible variable configurations.*

Definition 7.3.8: Admissible Variable Configuration *(cf. [Wilhelm, 2023])*

Let $\Sigma^f_{\mathcal{LA}}$ be a finite set of constraint representatives, and let $I \in \mathsf{Int}_c(\Sigma^f_{\mathcal{LA}})$ be a coherent interpretation. Then, we call

$$\mathsf{vconf}(I) = \mathsf{Sol}(\mathcal{LA}(I))$$

the set of the *variable configurations* which are *admissible* in $I \in \mathsf{Int}_c(\Sigma^f_{\mathcal{LA}})$.

By definition, coherent interpretations have at least one admissible variable configuration. With

$$\mathsf{vconf}(A) = \bigcup_{I \in \mathsf{Int}_c(\Sigma^f_{\mathcal{LA}}) \colon I \models A} \mathsf{vconf}(I)$$

we generalize the notion of admissible variable configurations to formulas $A \in \mathcal{L}(\Sigma^f_{\mathcal{LA}})$.

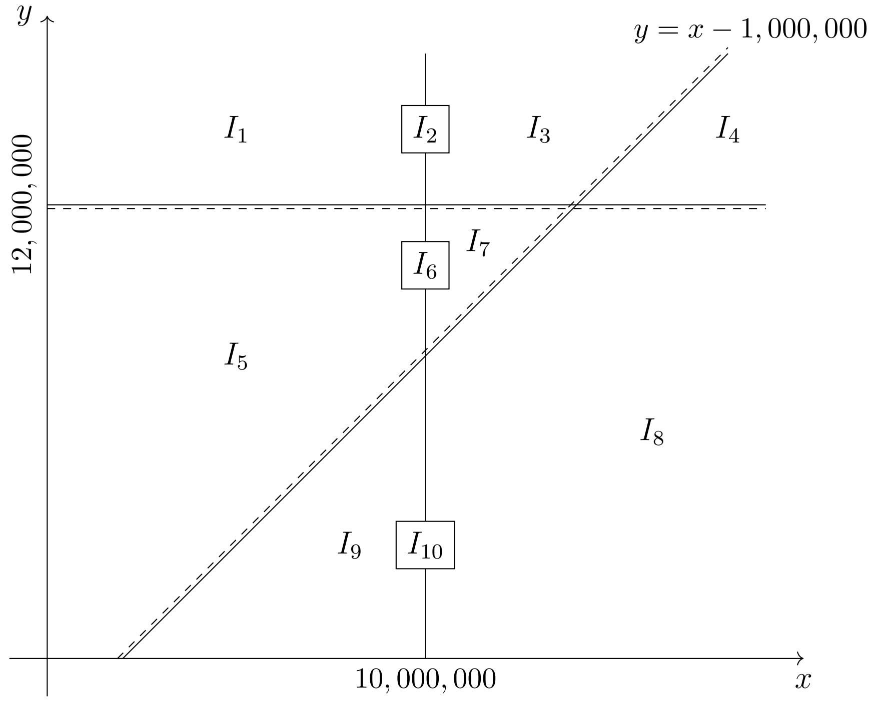

Figure 7.1: Areas of the admissible variable configurations of the coherent interpretations $I_1, \ldots, I_{10}$ from Example 7.3.9. Dashed lines indicate that the limit is still within the area.

Example 7.3.9: *(cf. [Wilhelm, 2023])*

We consider the following scenario: A shareholder wonders if the company in which she holds shares will pay a dividend at the end of the year. She believes with a probability of 0.8 that this is the case if the company makes high profit, i.e., if the income exceeds the expenses by more than €1,000,000. So far, the company has produced an income of €10,000,000. There are further payments pending, but it is unclear whether they will be executed this year. The expenses are unlikely to exceed €12,000,000 as the main investments have already been finalized. With a probability of 0.9 they will stay below the threshold of €12,000,000.

The arithmetic constraints mentioned in this example can be extracted from the text and formalized as follows:

$$\begin{aligned} p\colon &\quad & x - y >{}& \ 1,000,000, \\ r\colon &\quad & x ={}& 10,000,000, \end{aligned}$$

$$\begin{aligned} o: &\qquad x > 10,000,000, \\ e: &\qquad y \leq 12,000,000, \end{aligned}$$

where p stands for *high profit* ("The income x exceeds the expenses y by more than €1,000,000."), r for *received income* ("The received income is €10,000,000."), o for *income with outstanding payments* ("The income with outstanding payments is higher than €10,000,000."), and e for *expenses* ("The expenses are not higher than €12,000,000."). We assume that x and y are real numbers but the example would work with the assumption that x and y are integer numbers, too.[a] The corresponding constraint representatives are

$$\Sigma_{\mathcal{LA}}^{f,\mathsf{ex}} = \{a_p, a_r, a_o, a_e\}.$$

Not all interpretations in $\mathsf{Int}(\Sigma_{\mathcal{LA}}^{f,\mathsf{ex}})$ are coherent. For example, the constraints r and o cannot be satisfied jointly and, thus, interpretations $I \in \mathsf{Int}(\Sigma_{\mathcal{LA}}^{f,\mathsf{ex}})$ with $I(a_r) = I(a_o) = 1$ are not coherent. Likewise, interpretations with $I(a_p) = 1$ but $I(a_o) = I(a_e) = 0$ are not coherent because

$$\begin{aligned} \mathsf{Sol}(\{p, \widehat{o}, \widehat{e}\}) = \mathsf{Sol}(\{&x - y > 1,000,000, \\ &x \leq 10,000,000,\ y > 12,000,000\}) = \emptyset. \end{aligned}$$

The coherent interpretations in $\mathsf{Int}_c(\Sigma_{\mathcal{LA}}^{f,\mathsf{ex}})$, denoted as possible worlds, i.e., as complete conjunctions of those literals which are true in the interpretation, are

$$\begin{array}{lll} I_1 = \overline{a_p}\,\overline{a_r}\,\overline{a_o}\,\overline{a_e}, & I_2 = \overline{a_p}a_r\overline{a_o}\,\overline{a_e}, & I_3 = \overline{a_p}\,\overline{a_r}a_o\overline{a_e}, \\ I_4 = a_p\overline{a_r}a_o\overline{a_e}, & I_5 = \overline{a_p}\,\overline{a_r}\,\overline{a_o}a_e, & I_6 = \overline{a_p}a_r\overline{a_o}a_e, \\ I_7 = \overline{a_p}\,\overline{a_r}a_oa_e, & I_8 = a_p\overline{a_r}a_oa_e, & I_9 = a_p\overline{a_r}\,\overline{a_o}a_e, \\ I_{10} = a_pa_r\overline{a_o}a_e, & & \end{array}$$

For instance, we have (cf. also Example 7.3.4)

$$\mathsf{vconf}(a_pa_r\overline{a_o}a_e) = \{(x,y) \in \mathbb{Z}^2 \mid x = 10,000,000, \quad y < 9,000,000\}.$$

In Figure 7.1 it is visualized how the admissible variable configurations of the coherent interpretations $I_1, \ldots, I_{10}$ partition the Euclidean plane.

---

[a] Note that it would be plausible to assume that x and y are non-negative. We do not model this additional assumption in form of linear arithmetic constraints in order to keep the example simple, though.

Focusing on coherent interpretations yields a notion of *satisfiability modulo theory* (SMT, cf. [Barrett et al., 2021]).

Definition 7.3.10: Satisfiability Modulo $\mathcal{LA}$ *[Wilhelm, 2023]*

Let $\Sigma^f_{\mathcal{LA}}$ be a finite set of constraint representatives. A formula $A \in \mathcal{L}(\Sigma^f_{\mathcal{LA}})$ is *satisfiable modulo $\mathcal{LA}$* iff there is a coherent interpretation $I \in \mathsf{Int}_c(\Sigma^f_{\mathcal{LA}})$ with $I(A) = 1$.

The coherence of interpretations has a much bigger influence than just excluding self-contradictory interpretations from reasoning as the next proposition shows.

Proposition 7.3.11: Coherence and Variable Configurations *[Wilhelm, 2023]*

Let $\Sigma^f_{\mathcal{LA}}$ be a finite set of constraint representatives, and let $I, I' \in \mathsf{Int}_c(\Sigma^f_{\mathcal{LA}})$ be coherent interpretations with $I \neq I'$. Then, $\mathsf{vconf}(I) \cap \mathsf{vconf}(J) = \emptyset$ and thus, in particular, $\mathsf{vconf}(I) \neq \mathsf{vconf}(I')$.

Proof. Since $I \neq I'$, there is a constraint representative $a_c \in \Sigma^f_{\mathcal{LA}}$ with $I(a_c) \neq I'(a_c)$. Without loss of generality, we assume $I(a_c) = 1$ and $I'(a_c) = 0$. Then, $I(a_c) = 1$ implies $c \in \mathcal{LA}(I)$ and $\mathsf{vconf}(I) \subseteq \mathsf{Sol}(\{c\})$ follows. Note that the variables which are not mentioned in c but in $\mathcal{LA}(I)$ are not restricted by the solutions in $\mathsf{Sol}(\{c\})$ and can be chosen freely in order that $\mathsf{vconf}(I)$ and $\mathsf{Sol}(\{c\})$ refer to the same set of variables. On the other hand, $I'(a_c) = 0$ implies $\widehat{c} \in \mathcal{LA}(I')$ and $\mathsf{vconf}(I') \subseteq \mathsf{Sol}(\{\widehat{c}\})$ holds. Because $\mathsf{Sol}(\{c\}) \cap \mathsf{Sol}(\{\widehat{c}\}) = \emptyset$, we have $\mathsf{vconf}(I) \cap \mathsf{vconf}(I') \subseteq \mathsf{Sol}(\{c\}) \cap \mathsf{Sol}(\{\widehat{c}\}) = \emptyset$, too. Therefrom also the inequality $\mathsf{vconf}(I) \neq \mathsf{vconf}(I')$ follows. □

According to Proposition 7.3.11, coherent interpretations are unique semantic entities in the sense that different coherent interpretations refer to different sets of admissible variable configurations. Even the stronger statement that the sets of admissible variable configurations are disjoint holds, i.e., different coherent interpretations do not allow for the same variable configuration.

To summarize, the benefit of considering constraint representatives instead of simply encoding constraints as propositional atoms is that we achieve the ability to test the satisfiability of (sets of) constraint representatives modulo the theory of linear arithmetic. Therewith, we can interrelate constraint representatives like in $\neg a_c \equiv_C a_{\widehat{c}}$. Encoding constraints as atoms instead would undermine the fact that constraints are not semantically independent.

▶ Integrating Linear Arithmetic Constraints Into $\mathcal{ALC}^{\mathsf{ME}}$

Now, we extend the Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ by linear arithmetic constraints to the Description Logic $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$. For this, let $\mathcal{N}_I$, $\mathcal{N}_C$, and $\mathcal{N}_R$ be disjoint, finite sets of individual, concept, and role names such as in $\mathcal{ALC}^{\mathsf{ME}}$. Further, let $\Sigma^f_{\mathcal{LA}}$ be a finite set of constraint representatives which is also disjoint with the sets above. We use the constraint representatives from $\Sigma^f_{\mathcal{LA}}$ for a novel concept constructor.

Definition 7.3.12: $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-Concept
An $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-*concept* is either a concept name $A \in \mathcal{N}_C$ or of the form

$$\top, \quad \bot, \quad \neg C, \quad C \sqcup D, \quad C \sqcap D, \quad \exists r.A, \quad \forall r.A, \quad A_c,$$

where $A \in \mathcal{N}_C$ is a concept name, C and D are $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-concepts, $r \in \mathcal{N}_R$ is a role name, and $A_c \in \Sigma^f_{\mathcal{LA}}$ is a constraint representative.

In $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$ we can use the same concept constructors as in $\mathcal{ALC}^{\mathsf{ME}}$ (cf. Definition 7.1.1) as well as linear arithmetic constraints as novel concepts. The disallowance of complex concepts C in quantifications does not restrict the expressivity as we can replace $\exists r.C$ by $\exists r.A$, $C \sqsubseteq A$, and $A \sqsubseteq C$ in knowledge bases, and analogously for universal quantification, but ensures that linear arithmetic constraints do not occur in quantifications.

Note that we use uppercase letters for constraint representatives that are used as concepts in order to stick to the convention in Description Logics that concept symbols are uppercase letters and lowercase letters are reserved for role names and individual names. Informally spoken, a *linear arithmetic constraint concept* A_c represents those individuals for which the linear arithmetic constraint c is satisfied. For example, the concept A_c with $c\colon x_{\mathsf{height}} < 180$ could stand for all persons which are smaller than 180 centimeters.

All further syntactical notions such as the definitions of general concept inclusions and assertions carry over from $\mathcal{ALC}^{\mathsf{ME}}$ to $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$ by simply replacing $\mathcal{ALC}^{\mathsf{ME}}$-concepts through $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-concepts. For instance, assertions in $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$ are of the form $r(a, b)$ with $r \in \mathcal{N}_{\mathcal{R}}$ and $a, b \in \mathcal{N}_I$ as common, or of the form $C(a)$ where $a \in \mathcal{N}_I$ and C is an $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-concept including linear arithmetic constraint concepts. An $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-*knowledge base* is a triple $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ consisting of a TBox $\mathcal{T}$, an ABox $\mathcal{A}$, and a CBox $\mathcal{C}$, all defined with respect to $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-concepts. Conditionals in $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$ express probabilistic beliefs about qualitative logical statements combined with statements about admissible variable configurations.

Example 7.3.13

The beliefs of the shareholder from Example 7.3.9 can be formalized in the knowledge base $\mathcal{R}_{\mathsf{ex}} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ with $\mathcal{T} = \{\top \sqsubseteq A_r \sqcup A_o\}$, $\mathcal{A} = \emptyset$, and $\mathcal{C} = \{r_1, r_2\}$, where

$$r_1 = (D|A_p)[0.8], \quad r_2 = (A_e|\top)[0.9].$$

The meaning of the constraint representatives A_p, A_r, A_o, and A_e is discussed in Example 7.3.9. The additional concept name D stands for *d̲ividend is paid.*

To simplify our example and focus on the impact of the constraint representatives in $\mathcal{R}_{\mathsf{ex}}$, we assume that $|\mathcal{N}_I| = 1$ holds. This fits well to the scenario in the example when the individual name in $\mathcal{N}_I$ represents the company in which the shareholder holds shares.

The general concept inclusion in $\mathcal{T}$ states that the income of the company is €$10,000,000$ (A_r) or higher (A_o). Conditional r_1 states that if the profit of the company is high (A_p), then a dividend is paid (D) with probability 0.8, where the meaning of "high profit" is specified in the constraint p. Conditional r_2 states that the company's expenses do not exceed €$12,000,000$ with a probability of 0.9.

Also the semantics of $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$ is the same as the semantics of $\mathcal{ALC}^{\mathsf{ME}}$ albeit we have to clarify how arithmetic constraint concepts are interpreted.

Definition 7.3.14: Coherent $\mathcal{DL}$-Interpretation

Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a $\mathcal{DL}$-interpretation. We extend $\mathcal{I}$ to linear arithmetic constraint concepts $A_c \in \Sigma^f_{\mathcal{LA}}$ by mapping A_c to a subset $A_c^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$. An extended $\mathcal{DL}$-interpretation $\mathcal{I}$ is *coherent* iff for all individuals $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ it holds that

$$\mathcal{LA}(\mathcal{I}, a^{\mathcal{I}}) = \{c \mid A_c \in \Sigma^f_{\mathcal{LA}} \colon a^{\mathcal{I}} \in A_c^{\mathcal{I}}\} \cup \{\widehat{c} \mid A_c \in \Sigma^f_{\mathcal{LA}} \colon a^{\mathcal{I}} \notin A_c^{\mathcal{I}}\}$$

is satisfiable, i.e., $\mathsf{Sol}(\mathcal{LA}(\mathcal{I}, a^{\mathcal{I}})) \neq \emptyset$ holds.

Note that $\mathcal{LA}(\mathcal{I}, a^{\mathcal{I}})$ has an additional argument in contrast to $\mathcal{LA}(I)$ from (7.20) because the set of constraints $\mathcal{LA}(\mathcal{I}, a^{\mathcal{I}})$ can differ from individual to individual. Similar to the situation in $\mathcal{ALC}^{\mathsf{ME}}$, we assume that $\mathcal{N}_I$, $\mathcal{N}_C$, and $\mathcal{N}_R$ are finite sets and the unique name assumption holds. In particular, we fix the finite domain $\Delta = \mathcal{N}_I$. Further, we denote the set of all coherent $\mathcal{DL}$-interpretations with $\mathfrak{I}_c(\Sigma^f_{\mathcal{LA}}) = \mathfrak{I}_c(\mathcal{N}_I, \mathcal{N}_C, \mathcal{N}_R, \Sigma^f_{\mathcal{LA}})$. Note that we use the briefer notation $\mathfrak{I}_c(\Sigma^f_{\mathcal{LA}})$ as we do not aim at varying $\Delta = \mathcal{N}_I$, $\mathcal{N}_C$, or $\mathcal{N}_R$ in this section. If necessary, we will specify these sets at appropriate places.

ω	$\mathcal{P}^{\mathsf{ME}}_{\Sigma^f_{\mathcal{LA}},\mathcal{R}_{\mathsf{ex}}}(\omega)$	ω	$\mathcal{P}^{\mathsf{ME}}_{\Sigma^f_{\mathcal{LA}},\mathcal{R}_{\mathsf{ex}}}(\omega)$
$da_pa_r\overline{a}_oa_e$	0.166	$\bar{d}a_pa_r\overline{a}_oa_e$	0.042
$da_p\overline{a}_ra_oa_e$	0.166	$\bar{d}a_p\overline{a}_ra_oa_e$	0.042
$da_p\overline{a}_ra_o\overline{a}_e$	0.018	$\bar{d}a_p\overline{a}_ra_o\overline{a}_e$	0.005
$da_p\overline{a}_r\overline{a}_oa_e$	0	$\bar{d}a_p\overline{a}_r\overline{a}_oa_e$	0
$d\overline{a}_pa_r\overline{a}_oa_e$	0.126	$\bar{d}\overline{a}_pa_r\overline{a}_oa_e$	0.126
$d\overline{a}_pa_r\overline{a}_o\overline{a}_e$	0.014	$\bar{d}\overline{a}_pa_r\overline{a}_o\overline{a}_e$	0.014
$d\overline{a}_p\overline{a}_ra_oa_e$	0.126	$\bar{d}\overline{a}_p\overline{a}_ra_oa_e$	0.126
$d\overline{a}_p\overline{a}_ra_o\overline{a}_e$	0.014	$\bar{d}\overline{a}_p\overline{a}_ra_o\overline{a}_e$	0.014
$d\overline{a}_p\overline{a}_r\overline{a}_oa_e$	0	$\bar{d}\overline{a}_p\overline{a}_r\overline{a}_oa_e$	0
$d\overline{a}_p\overline{a}_r\overline{a}_o\overline{a}_e$	0	$\bar{d}\overline{a}_p\overline{a}_r\overline{a}_o\overline{a}_e$	0

Table 7.3: MaxEnt model $\mathcal{P}^{\mathsf{ME}}_{\Sigma^f_{\mathcal{LA}},\mathcal{R}_{\mathsf{ex}}}$ of the knowledge base $\mathcal{R}_{\mathsf{ex}}$ from Example 7.3.13. The probabilities are rounded to three decimal places.

The definition of probabilistic models of $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-knowledge bases is the same as in $\mathcal{ALC}^{\mathsf{ME}}$ albeit we consider coherent interpretations only. If an $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-knowledge base has a (positive) probabilistic model, then we call it consistent (p-consistent). Also the definition of the MaxEnt model $\mathcal{P}^{\mathsf{ME}}_{\Sigma^f_{\mathcal{LA}},\mathcal{R}}$ carries over then. Note that the probability $\mathcal{P}^{\mathsf{ME}}_{\Sigma^f_{\mathcal{LA}},\mathcal{R}}(A_c(a))$ of the constraint representative A_c associated to $a \in \mathcal{N}_I$ reflects the subjective belief of a MaxEnt reasoner with knowledge base $\mathcal{R}$ in the satisfaction of the constraint c for a. It does not depend on a measure on the possible variable configurations. In particular, it does not depend on the number of solutions in $\mathsf{Sol}(\{c\})$.

Example 7.3.15

We recall the knowledge base $\mathcal{R}_{\mathsf{ex}}$ from Example 7.3.13, as well as $\mathcal{N}_I = \{c\}$ and the set $\Sigma^f_{\mathcal{LA}} = \{a_p, a_r, a_o, a_e\}$ of constraint representatives which are used in $\mathcal{R}_{\mathsf{ex}}$. There are 20 coherent $\mathcal{DL}$-interpretations in $\mathfrak{I}_c(\Sigma^f_{\mathcal{LA}})$ (cf. Example 7.3.9 and the fact that either $c^{\mathcal{I}} \in D^{\mathcal{I}}$ holds or not). The MaxEnt model $\mathcal{P}^{\mathsf{ME}}_{\Sigma^f_{\mathcal{LA}},\mathcal{R}_{\mathsf{ex}}}$ of the knowledge base $\mathcal{R}_{\mathsf{ex}}$ from Example 7.3.13 is shown in Table 7.3. For a better readability, the $\mathcal{DL}$-interpretations are written as possible worlds. For example, for the only constant $c \in \mathcal{N}_I$, the company under consideration, the $\mathcal{DL}$-interpretation with $c^{\mathcal{I}} \in D^{\mathcal{I}}, A_p^{\mathcal{I}}, A_r^{\mathcal{I}}, A_e^{\mathcal{I}}$ and $c^{\mathcal{I}} \notin A_o^{\mathcal{I}}$ is represented as $da_pa_r\overline{a}_oa_e$.

We can (approximatively) MaxEnt-infer the conditional $(D|\top)[0.630]$

from $\mathcal{R}_{\mathsf{ex}}$. That is, following the MaxEnt principle and the shareholder's beliefs as formalized in $\mathcal{R}_{\mathsf{ex}}$, the shareholder should believe in the payout of a dividend with probability 0.630.

Unfortunately, reasoning in $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$ depends on the selection of the constraints which are represented in $\Sigma^f_{\mathcal{LA}}$ as we will see next. However, we will show how to overcome this and draw inferences that are independent of this selection and, therewith, of the syntactical representation of admissible variable configurations. Actually, we will discuss an enhanced maximum entropy inference relation which allows us to infer probability bounds for constraints that are not even represented in $\Sigma^f_{\mathcal{LA}}$, too.

▶ Becoming Independent of the Syntactical Representation of Variable Configurations

Once a set $\Sigma^f_{\mathcal{LA}}$ of constraint representatives is fixed, MaxEnt reasoning in $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$ is independent of the syntactical representation of admissible variable configurations because the coherent interpretations in $\mathfrak{I}_c(\Sigma^f_{\mathcal{LA}})$ are semantically unique entities which differ in their variable configurations (cf. Proposition 7.3.11). However, in $\mathcal{LA}$ it is possible to specify one and the same solution set with different constraint sets and, hence, the selection of the constraint representatives in $\Sigma^f_{\mathcal{LA}}$ is ambiguous. For instance, the combination of the constraints $x \leq 10,000,000$ and $x \geq 10,000,000$ determines the same solution $x = 10,000,000$ as the constraint $x = 10,000,000$ itself. Accordingly, the choice of $\Sigma^f_{\mathcal{LA}}$ has a considerable impact on the MaxEnt model of a knowledge base in $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$.

Example 7.3.16

In $\mathcal{R}_{\mathsf{ex}}$ from Example 7.3.13 the general concept inclusion $\top \sqsubseteq A_r \sqcup A_o$ occurs which states that the company's income is equal to or higher than € 10, 000, 000. With the constraint representative A_t, which refers to the constraint

$$t\colon \qquad x \geq 10,000,000,$$

the GCI $\top \sqsubseteq A_r \sqcup A_o$ can be rewritten to $\top \sqsubseteq A_t$ without altering its meaning. We denote the resulting knowledge base with $\mathcal{R}_{ex2}$. The maximum entropy model $\mathcal{P}^{\mathsf{ME}}_{\Sigma^{f'}_{\mathcal{LA}},\mathcal{R}_{\mathsf{ex2}}}$ of $\mathcal{R}_{\mathsf{ex2}}$ with respect to the new set of constraint representatives $\Sigma^{f'}_{\mathcal{LA}}$, which mentions A_t instead of A_r and A_o, differs from $\mathcal{P}^{\mathsf{ME}}_{\Sigma^f_{\mathcal{LA}},\mathcal{R}^{\mathsf{ex}}}$ not only because the number of coherent interpretations differs. Also, the inferred probabilities differ. For example, from $\mathcal{R}_{\mathsf{ex2}}$ the conditional $(D|\top)[p]$ can be inferred with a MaxEnt probability $p \approx 0.635$. When the same conditional is inferred from $\mathcal{R}^{\mathsf{ex}}$, the MaxEnt probability is $p \approx 0.630$ instead (cf. Example 7.3.15).

Now, one could ask if there is a preferred selection of constraint representatives for reasoning in $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$. A common way of determining the formal language in which reasoning takes place is to choose the language which is induced by the knowledge base. This, however, delegates the responsibility of a proper selection of constraint representatives to the knowledge engineer only. If this selection is purely pragmatic and without a profound justification, one probably would like to become independent of it now as before. For this reason, we show how it is possible to completely abstract from the syntactical representation of admissible variable configurations in $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$. The basic idea is to aggregate coherent $\mathcal{DL}$-interpretations to equivalence classes and define probabilistic models over these equivalence classes instead of the $\mathcal{DL}$-interpretations directly. Because we define the equivalence classes semantically wrt. the evaluation of $\mathcal{R}$, reasoning becomes independent of the syntax, particularly of the selection of the constraint representatives in $\Sigma^{f}_{\mathcal{LA}}$.

Definition 7.3.17: $\mathcal{R}$-Equivalence

Let $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ be an $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-knowledge base, and let $\mathcal{I}, \mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f}_{\mathcal{LA}})$ be coherent $\mathcal{DL}$-interpretations. We call $\mathcal{I}$ and $\mathcal{I}'$ *$\mathcal{R}$-equivalent* iff both

- $\mathcal{I}$ and $\mathcal{I}'$ are *logically identical*, i.e., $a^{\mathcal{I}} = a^{\mathcal{I}'}$ for all $a \in \mathcal{N}_I$,[a] $C^{\mathcal{I}} = C^{\mathcal{I}'}$ for all $C \in \mathcal{N}_C$, and $r^{\mathcal{I}} = r^{\mathcal{I}'}$ for all $r \in \mathcal{N}_R$ holds,
- (1) both $\mathcal{I}$ and $\mathcal{I}'$ dissatisfy *some* general concept inclusions from $\mathcal{T}$ or assertions from $\mathcal{A}$, or (2) both $\mathcal{I}$ and $\mathcal{I}'$ satisfy $\mathcal{T}$ and $\mathcal{A}$, and for all $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-conditionals $(D|C)[p] \in \mathcal{C}$ it holds that both $(C \sqcap D)^{\mathcal{I}} = (C \sqcap D)^{\mathcal{I}'}$ and $(C \sqcap \neg D)^{\mathcal{I}} = (C \sqcap \neg D)^{\mathcal{I}'}$.

If $\mathcal{I}$ and $\mathcal{I}'$ are $\mathcal{R}$-equivalent, we write $\mathcal{I} \sim_{\mathcal{R}} \mathcal{I}'$. $\mathcal{R}$-equivalence constitutes an equivalence relation between coherent $\mathcal{DL}$-interpretations, and we denote the respective equivalence classes with

$$[\mathcal{I}]_{\mathcal{R}} = \{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f}_{\mathcal{LA}}) \mid \mathcal{I} \sim_{\mathcal{R}} \mathcal{I}'\}.$$

The set of all equivalence classes is denoted by $\mathfrak{I}^{\sim}_{\mathcal{R}} = \{[\mathcal{I}]_{\mathcal{R}} \mid \mathcal{I} \in \mathfrak{I}_c(\Sigma^{f}_{\mathcal{LA}})\}$. Further, we use the notations $[\mathcal{I}]_{\mathcal{R}} \models \mathcal{T}$ and $[\mathcal{I}]_{\mathcal{R}} \models \mathcal{A}$ iff $\mathcal{I} \models \mathcal{T}$ respectively iff $\mathcal{I} \models \mathcal{A}$.

---

[a] Note that we assume $a^{\mathcal{I}} = a^{\mathcal{I}'}$ for all $a \in \mathcal{N}_I$ anyway.

Informally speaking, equivalent coherent $\mathcal{DL}$-interpretations evaluate individual names, concept names, and role names the same. Apart from this, they may dif-

$[\omega_i]_{\mathcal{R}_{\mathsf{ex}}}$	$\widetilde{\mathcal{P}}^{\mathsf{ME}}_{\mathcal{R}_{\mathsf{ex}}}([\omega_i]_{\mathcal{R}_{\mathsf{ex}}})$
$[\omega_1]_{\mathcal{R}_{\mathsf{ex}}} := \{d\overline{a_p}\,\overline{a_r}\,\overline{a_o}\,\overline{a_e}\}$	0
$[\omega_2]_{\mathcal{R}_{\mathsf{ex}}} := \{d\overline{a_p}a_r\overline{a_o}\,\overline{a_e},\ d\overline{a_p}\,\overline{a_r}a_o\overline{a_e}\}$	0.027
$[\omega_3]_{\mathcal{R}_{\mathsf{ex}}} := \{da_p\overline{a_r}a_o\overline{a_e}\}$	0.036
$[\omega_4]_{\mathcal{R}_{\mathsf{ex}}} := \{d\overline{a_p}\,\overline{a_r}\,\overline{a_o}a_e\}$	0
$[\omega_5]_{\mathcal{R}_{\mathsf{ex}}} := \{d\overline{a_p}a_r\overline{a_o}a_e,\ d\overline{a_p}\,\overline{a_r}a_oa_e\}$	0.247
$[\omega_6]_{\mathcal{R}_{\mathsf{ex}}} := \{da_p\overline{a_r}\,\overline{a_o}a_e\}$	0
$[\omega_7]_{\mathcal{R}_{\mathsf{ex}}} := \{da_pa_r\overline{a_o}a_e,\ da_p\overline{a_r}a_oa_e\}$	0.325
$[\omega_8]_{\mathcal{R}_{\mathsf{ex}}} := \{\bar{d}\overline{a_p}\,\overline{a_r}\,\overline{a_o}\,\overline{a_e}\}$	0
$[\omega_9]_{\mathcal{R}_{\mathsf{ex}}} := \{\bar{d}\overline{a_p}a_r\overline{a_o}\,\overline{a_e},\ \bar{d}\overline{a_p}\,\overline{a_r}a_o\overline{a_e}\}$	0.027
$[\omega_{10}]_{\mathcal{R}_{\mathsf{ex}}} := \{\bar{d}a_p\overline{a_r}a_o\overline{a_e}\}$	0.009
$[\omega_{11}]_{\mathcal{R}_{\mathsf{ex}}} := \{\bar{d}\overline{a_p}\,\overline{a_r}\,\overline{a_o}a_e\}$	0
$[\omega_{12}]_{\mathcal{R}_{\mathsf{ex}}} := \{\bar{d}\overline{a_p}a_r\overline{a_o}a_e,\ \bar{d}\overline{a_p}\,\overline{a_r}a_oa_e\}$	0.247
$[\omega_{13}]_{\mathcal{R}_{\mathsf{ex}}} := \{\bar{d}a_p\overline{a_r}\,\overline{a_o}a_e\}$	0
$[\omega_{14}]_{\mathcal{R}_{\mathsf{ex}}} := \{\bar{d}a_pa_r\overline{a_o}a_e,\ \bar{d}a_p\overline{a_r}a_oa_e\}$	0.081

Table 7.4: $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-equivalence classes of coherent $\mathcal{DL}$-interpretations in $\mathfrak{I}^{\sim}_{\mathcal{R}_{\mathsf{ex}}}$ with respect to $\mathcal{R}_{\mathsf{ex}}$ from Example 7.3.13 (denoted as possible worlds) and their aggregated MaxEnt probabilities. The probabilities are rounded to three decimal places.

fer in the evaluation of constraint representatives as long as this does not affect the evaluation of the knowledge base $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$. In particular, the verification and the falsification of the conditionals in $\mathcal{C}$ has to be the same. Hence, an equivalence class $[\mathcal{I}]_{\mathcal{R}}$ can be understood as the specification on how to interpret individual, concept, and role names together with the set of admissible variable configurations $\mathsf{vconf}_a([\mathcal{I}]_{\mathcal{R}})$ for the individuals $a \in \mathcal{N}_I$, where $\mathsf{vconf}_a([\mathcal{I}]_{\mathcal{R}})$ is given by

$$\mathsf{vconf}_a([\mathcal{I}]_{\mathcal{R}}) = \bigcup_{\mathcal{I}' \in [\mathcal{I}]_{\mathcal{R}}} \mathsf{vconf}_a(\mathcal{I}')$$

and $\mathsf{vconf}_a(\mathcal{I}) = \mathsf{Sol}(\mathcal{LA}(\mathcal{I}, a^{\mathcal{I}}))$.

Example 7.3.18

We recall $\mathcal{R}_{\mathsf{ex}}$ from Example 7.3.13. The $\mathcal{R}_{\mathsf{ex}}$-equivalence classes are shown in Table 7.4 (for an explanation of the notations see Example 7.3.15). The equivalence classes aggregate coherent $\mathcal{DL}$-interpretations such that the two constraints $r\colon x = 10,000,000$ and $o\colon x > 10,000,000$ are combined. For example, we have

$$\begin{aligned}\mathsf{vconf}_c([\omega_7]_{\mathcal{R}_{\mathsf{ex}}}) &= \mathsf{Sol}(\{p,\ x = 10,000,000,\ e\}) \cup \mathsf{Sol}(\{p,\ x > 10,000,000,\ e\})\\ &= \mathsf{Sol}(\{p,\ x \geq 10,000,000,\ e\}),\end{aligned}$$

where c is the only individual name in $\mathcal{N}_I$ representing the considered company. The aggregation is possible because the differentiation between the constraints r and o is irrelevant for the evaluation of $\mathcal{R}_{\mathsf{ex}}$, and it matches the idea of replacing $A_r \sqcup A_o$ by A_t as proposed in Example 7.3.16.

Now, we discuss how probabilistic reasoning in general and MaxEnt reasoning in particular works when considering equivalence classes of coherent $\mathcal{DL}$-interpretations instead of the $\mathcal{DL}$-interpretations themselves.

Definition 7.3.19: Aggregated (Maximum Entropy) Model

Let $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ be a consistent $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-knowledge base. A probability distribution $\widetilde{\mathcal{P}}\colon \mathfrak{I}^{\sim}_{\mathcal{R}} \to [0,1]$ is an *aggregated model* of $\mathcal{R}$ iff

1. for all $C \sqsubseteq D \in \mathcal{T}$ and $[\mathcal{I}]_{\mathcal{R}} \in \mathfrak{I}^{\sim}_{\mathcal{R}}$,

$$\mathcal{I} \not\models (C \sqsubseteq D) \quad \text{implies} \quad \widetilde{\mathcal{P}}([\mathcal{I}]_{\mathcal{R}}) = 0,$$

2a. for all assertions $A(a) \in \mathcal{A}$ and $[\mathcal{I}]_{\mathcal{R}} \in \mathfrak{I}^{\sim}_{\mathcal{R}}$,

$$\mathcal{I} \not\models A(a) \quad \text{implies} \quad \widetilde{\mathcal{P}}([\mathcal{I}]_{\mathcal{R}}) = 0,$$

2b. for all assertions $r(a,b) \in \mathcal{A}$ and $[\mathcal{I}]_{\mathcal{R}} \in \mathfrak{I}^{\sim}_{\mathcal{R}}$,

$$\mathcal{I} \not\models r(a,b) \quad \text{implies} \quad \widetilde{\mathcal{P}}([\mathcal{I}]_{\mathcal{R}}) = 0,$$

3. for all conditionals $(D|C)[p] \in \mathcal{C}$,

$$\sum_{[\mathcal{I}]_{\mathcal{R}} \models \mathcal{T},\mathcal{A}} \left(|(C \sqcap D)^{\mathcal{I}}| + |(C \sqcap \neg D)^{\mathcal{I}}|\right) \cdot \widetilde{\mathcal{P}}([\mathcal{I}]_{\mathcal{R}}) > 0$$

$$\text{and} \quad \frac{\sum_{[\mathcal{I}]_{\mathcal{R}} \models \mathcal{T},\mathcal{A}} |(C \sqcap D)^{\mathcal{I}}| \cdot \widetilde{\mathcal{P}}([\mathcal{I}]_{\mathcal{R}})}{\sum_{[\mathcal{I}]_{\mathcal{R}} \models \mathcal{T},\mathcal{A}} (|(C \sqcap D)^{\mathcal{I}}| + |(C \sqcap \neg D)^{\mathcal{I}}|) \cdot \widetilde{\mathcal{P}}([\mathcal{I}]_{\mathcal{R}})} = p. \quad (7.21)$$

If $\widetilde{\mathcal{P}}$ is an aggregated model of $\mathcal{R}$, then we write $\widetilde{\mathcal{P}} \models^{\sim} \mathcal{R}$. The *aggregated MaxEnt-model* of a consistent $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-knowledge base $\mathcal{R}$ is defined by

$$\widetilde{\mathcal{P}}^{\mathsf{ME}}_{\Sigma^f_{\mathcal{LA}},\mathcal{R}} = \arg \max_{\widetilde{\mathcal{P}} \models^{\sim} \mathcal{R}} - \sum_{[\mathcal{I}]_{\mathcal{R}} \in \mathfrak{I}^{\sim}_{\mathcal{R}}} \widetilde{\mathcal{P}}([\mathcal{I}]_{\mathcal{R}}) \cdot \log \widetilde{\mathcal{P}}([\mathcal{I}]_{\mathcal{R}}),$$

where the convention $0 \cdot \log 0 = 0$ applies.

Note that aggregated models are well-defined because the $\mathcal{R}$-equivalence classes are defined in such a way that the $\mathcal{DL}$-interpretations in a single equivalence class agree on the terms which are used in the formulas in Definition 7.3.19. For example, in (7.21) the counts $|C \sqcap D)^{\mathcal{I}}|$ and $|C \sqcap D)^{\mathcal{I}}|$ occur which are the same for all $\mathcal{DL}$-interpretations in $[\mathcal{I}]_{\mathcal{R}}$.

The central result in this subsection is that p-consistent $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-knowledge bases which differ at most in their syntactical representation of admissible variable configurations have the same aggregated maximum entropy model. For this, we adapt the concept of conditional structures for sets of possible worlds from Definition 3.2.3. For an $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-knowledge base $\mathcal{R} = (\mathcal{T}, \mathcal{A}, \mathcal{C})$ with $\mathcal{C} = \{(D_i|C_i)[p_i] \mid i = 1, \dots, n\}$, we define the conditional structure with respect to the set of coherent $\mathcal{DL}$-interpretations $\mathfrak{I}_c(\Sigma^f_{\mathcal{LA}})$ by

$$\sigma_{\mathcal{R}}(\mathfrak{I}_c(\Sigma^f_{\mathcal{LA}})) = \sum_{\mathcal{I} \in \mathfrak{I}_c(\Sigma^f_{\mathcal{LA}})} \sigma_{\mathcal{R}}(\mathcal{I})$$

where

$$\sigma_{\mathcal{R}}(\mathcal{I}) = \begin{cases} \prod_{i=1}^{n} \prod_{\substack{a \in \mathcal{N}_I : \\ a^{\mathcal{I}} \in (C_i \sqcap D_i)^{\mathcal{I}}}} \mathsf{a}_i^{+} \cdot \prod_{\substack{a \in \mathcal{N}_I : \\ a^{\mathcal{I}} \in (C_i \sqcap \neg D_i)^{\mathcal{I}}}} \mathsf{a}_i^{-} & \text{if} \quad \mathcal{I} \models \mathcal{T}, \mathcal{A} \\ 0 & \text{otherwise} \end{cases}.$$

That is, $\sigma_{\mathcal{R}}(\mathfrak{I}_c(\Sigma^f_{\mathcal{LA}}))$ aggregates the conditional structures of all coherent $\mathcal{DL}$-interpretations which satisfy the TBox $\mathcal{T}$ and the ABox $\mathcal{A}$ of $\mathcal{R}$. For the remaining coherent $\mathcal{DL}$-interpretations, the conditional structures do not play a role in respect of the (aggregated) maximum entropy model, because they have a MaxEnt probability of 0 by the definition of (aggregated) probabilistic models, and are set to 0.

Proposition 7.3.20: Syntax Independence of Aggregated MaxEnt Model
Let $\mathcal{R} = (\mathcal{T},\mathcal{A},\mathcal{C})$ and $\mathcal{R}' = (\mathcal{T}',\mathcal{A}',\mathcal{C}')$ be p-consistent $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-knowledge bases defined over $\mathcal{N}_I$, $\mathcal{N}_C$, and $\mathcal{N}_R$, as well as the finite sets of constraint representatives $\Sigma^{f}_{\mathcal{LA}}$ and $\Sigma^{f'}_{\mathcal{LA}}$, respectively. Further, let $|\mathcal{C}| = |\mathcal{C}'|$, and the conditionals in $\mathcal{C}$ and $\mathcal{C}'$ shall have the same probabilities. If $\sigma_{\mathcal{R}}(\mathfrak{I}_c(\Sigma^{f}_{\mathcal{LA}})) = \sigma_{\mathcal{R}'}(\mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}}))$, then there is a bijection $\beta\colon \mathfrak{I}^{\sim}_{\mathcal{R}} \to \mathfrak{I}'^{\sim}_{\mathcal{R}'}$ between the induced equivalence classes of coherent $\mathcal{DL}$-interpretations with $\beta([\mathcal{I}]_{\mathcal{R}}) = [\mathcal{I}']_{\mathcal{R}'}$ iff $\sigma_{\mathcal{R}}(\mathcal{I}) = \sigma_{\mathcal{R}'}(\mathcal{I}')$ such that for all $[\mathcal{I}]_{\mathcal{R}} \in \mathfrak{I}^{\sim}_{\mathcal{R}}$ it holds that

$$\widetilde{\mathcal{P}}^{\mathsf{ME}}_{\Sigma^{f}_{\mathcal{LA}},\mathcal{R}}([\mathcal{I}]_{\mathcal{R}}) = \widetilde{\mathcal{P}}^{\mathsf{ME}}_{\Sigma^{f'}_{\mathcal{LA}},\mathcal{R}'}(\beta([\mathcal{I}]_{\mathcal{R}})).$$

Proof. First, we note that $|\mathfrak{I}^{\sim}_{\mathcal{R}}| = |\mathfrak{I}^{\sim}_{\mathcal{R}'}|$ holds by construction. Further, with $\beta([\mathcal{I}]_{\mathcal{R}}) = [\mathcal{I}']_{\mathcal{R}'}$ iff $\sigma_{\mathcal{R}}(\mathcal{I}) = \sigma_{\mathcal{R}'}(\mathcal{I}')$ and the fact that the probabilities in $\mathcal{C}$ and $\mathcal{C}'$ are the same, it follows that for all $[\mathcal{I}]_{\mathcal{R}} \in \mathfrak{I}^{\sim}_{\mathcal{R}}$ we have $f_i(\mathcal{I}) = f_i(\mathcal{I}')$ for $\mathcal{I} \in [\mathcal{I}]_{\mathcal{R}}$ and $\mathcal{I}' \in \beta([\mathcal{I}]_{\mathcal{R}})$, where

$$f_i(\mathcal{I}) = |(C_i \sqcap D_i)^{\mathcal{I}}| \cdot (1 - p_i) - |(C_i \sqcap \neg D_i)^{\mathcal{I}}| \cdot p_i$$

is the *feature function* value of $\mathcal{I}$ with respect to the i-th conditional in $\mathcal{C}$ (cf. Definition 3.4.3). Therewith, the aggregated MaxEnt-models with respect to $\mathcal{R}$ and $\mathcal{R}'$ yield the product representations (cf. 4.13)

$$\widetilde{\mathcal{P}}^{\mathsf{ME}}_{\Sigma^{f}_{\mathcal{LA}},\mathcal{R}}([\mathcal{I}]_{\mathcal{R}}) = \begin{cases} \alpha_0 \cdot \prod_{i=1}^{n} \alpha_i^{f_i(\mathcal{I})} & \text{if} \quad \mathcal{I} \models \mathcal{T},\mathcal{A} \\ 0 & \text{otherwise} \end{cases},$$

$$\widetilde{\mathcal{P}}^{\mathsf{ME}}_{\Sigma^{f'}_{\mathcal{LA}},\mathcal{R}'}([\mathcal{I}]_{\mathcal{R}'}) = \begin{cases} \alpha_0 \cdot \prod_{i=1}^{n} \alpha_i^{f_i(\mathcal{I})} & \text{if} \quad \mathcal{I} \models \mathcal{T},\mathcal{A} \\ 0 & \text{otherwise} \end{cases}.$$

The alpha-values are of the same number, because $\mathcal{R}$ and $\mathcal{R}'$ mention the same number of conditionals. Further, they also have the same value because the feature function values are the same for $\mathcal{R}$ and $\mathcal{R}'$. Consequently,

$$\widetilde{\mathcal{P}}^{\mathsf{ME}}_{\Sigma^{f}_{\mathcal{LA}},\mathcal{R}}([\mathcal{I}]_{\mathcal{R}}) = \widetilde{\mathcal{P}}^{\mathsf{ME}}_{\Sigma^{f'}_{\mathcal{LA}},\mathcal{R}'}(\beta([\mathcal{I}]_{\mathcal{R}}))$$

for all $[\mathcal{I}]_{\mathcal{R}} \in \mathfrak{I}^{\sim}_{\mathcal{R}}$ holds. □

Proposition 7.3.20 proves that the aggregated MaxEnt-model is independent of the selection of the constraint representatives and we may write $\widetilde{\mathcal{P}}^{\mathsf{ME}}_{\mathcal{R}}$ instead of $\widetilde{\mathcal{P}}^{\mathsf{ME}}_{\Sigma^{f}_{\mathcal{LA}},\mathcal{R}}$.

Example 7.3.21

The aggregated MaxEnt-model $\widetilde{\mathcal{P}}^{\mathsf{ME}}_{\mathcal{R}_{\mathsf{ex}}}$ of the belief base $\mathcal{R}_{\mathsf{ex}}$ from Example 7.3.13 is shown in Table 7.4.

A drawback of aggregated models is that they do not provide a probability assignment to single coherent $\mathcal{DL}$-interpretations. Next, we tackle this problem and extend maximum entropy inferences such that they assign lower and upper probability bounds to coherent $\mathcal{DL}$-interpretations which are derived from the aggregated MaxEnt model. This extension is also capable of assigning probabilities to expressions involving constraint representatives which are not represented in $\Sigma^f_{\mathcal{LA}}$ so that one is not limited to the syntax of the $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-knowledge base when formulating queries.

▶ Drawing Inferences From the Aggregated Maximum Entropy Model

When drawing inferences from consistent $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-knowledge bases $\mathcal{R}$ based on their aggregated maximum entropy model $\widetilde{\mathcal{P}}^{\mathsf{ME}}_{\mathcal{R}}$, one is limited to ask queries that can be answered based on the probability assignment to the equivalence classes of the coherent $\mathcal{DL}$-interpretations in $\mathfrak{I}^{\sim}_{\mathcal{R}}$ in the first instance, because no probabilities are assigned to single $\mathcal{DL}$-interpretations. We overcome this limitation by computing lower and upper bounds for the maximum entropy probabilities of coherent $\mathcal{DL}$-interpretations in $\mathfrak{I}_c(\Sigma^f_{\mathcal{LA}})$. Furthermore, we apply this assignment to coherent $\mathcal{DL}$-interpretations which are defined with respect to any finite set of constraint representatives $\Sigma^{f'}_{\mathcal{LA}}$ that may differ from $\Sigma^f_{\mathcal{LA}}$ as well. To do so, we have to relate coherent $\mathcal{DL}$-interpretations from $\mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})$ to the equivalence classes in $\mathfrak{I}^{\sim}_{\mathcal{R}}$ which brings us to the notions of *conflict-free* and *refining (equivalence-classes of) interpretations.*

Definition 7.3.22: Conflict-free and Refining $\mathcal{DL}$-Interpretations

Let $\Sigma^f_{\mathcal{LA}}$ and $\Sigma^{f'}_{\mathcal{LA}}$ be finite sets of constraints, and let $I \in \mathfrak{I}_c(\Sigma^f_{\mathcal{LA}})$ and $I' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})$ be coherent $\mathcal{DL}$-interpretations. Then, we say that

- $\mathcal{I}$ is *conflict-free* with $\mathcal{I}'$ iff $\mathcal{I}$ and $\mathcal{I}'$ are logically identical (cf. Definition 7.3.17) and

$$\mathsf{vconf}_a(\mathcal{I}) \cap \mathsf{vconf}_a(\mathcal{I}') \neq \emptyset, \qquad a \in \mathcal{N}_I, \tag{7.22}$$

- $\mathcal{I}$ *refines* $\mathcal{I}'$ iff $\mathcal{I}$ and $\mathcal{I}'$ are logically identical and

$$\mathsf{vconf}_a(\mathcal{I}) \subseteq \mathsf{vconf}_a(\mathcal{I}'), \qquad a \in \mathcal{N}_I. \tag{7.23}$$

The conflict-freeness of two coherent $\mathcal{DL}$-interpretations $\mathcal{I}$ and $\mathcal{I}'$ is a symmetric property and means that they share an admissible variable configuration. If $\mathcal{I}$ refines $\mathcal{I}'$, then every admissible variable configuration of $\mathcal{I}$ is also an admissible variable configuration of $\mathcal{I}'$. Note that if $\Sigma^f_{\mathcal{LA}} = \Sigma^{f'}_{\mathcal{LA}}$, then the notions of conflict-free and refining $\mathcal{DL}$-interpretations become trivial.

Proposition 7.3.23: Conflict-free and Refining $\mathcal{DL}$-Interpretations

Let $\mathcal{I}, \mathcal{I}' \in \mathfrak{I}_c(\Sigma^f_{\mathcal{LA}})$ be coherent interpretations over the same set of constraint representatives $\Sigma^f_{\mathcal{LA}}$. If $\mathcal{I}$ is conflict-free with $\mathcal{I}'$ or $\mathcal{I}$ refines $\mathcal{I}'$, then $\mathcal{I} = \mathcal{I}'$.

Proof. In both cases, if $\mathcal{I}$ is conflict-free with $\mathcal{I}'$ or if $\mathcal{I}$ refines $\mathcal{I}'$, then $\mathcal{I}$ and $\mathcal{I}'$ are logically identical. It remains to show that $\mathcal{I}$ and $\mathcal{I}'$ agree on $\Sigma^f_{\mathcal{LA}}$. If $\mathcal{I}$ is conflict-free with $\mathcal{I}'$, then $\mathsf{vconf}_a(\mathcal{I}) \cap \mathsf{vconf}_a(\mathcal{I}') \neq \emptyset$ for $a \in \mathcal{N}_I$ holds. And if $\mathcal{I}$ refines $\mathcal{I}'$, then $\mathsf{vconf}_a(\mathcal{I}) \subseteq \mathsf{vconf}_a(\mathcal{I}')$ for $a \in \mathcal{N}_I$ holds. In both cases, $\mathsf{vconf}_a(\mathcal{I}) \cap \mathsf{vconf}_a(\mathcal{I}') \neq \emptyset$ for $a \in \mathcal{N}_I$ is true. Because Proposition 7.3.11 can be lifted to coherent $\mathcal{DL}$-interpretations in the sense that $\mathcal{I} \neq \mathcal{I}'$ implies $\mathsf{vconf}_a(\mathcal{I}) \cap \mathsf{vconf}_a(\mathcal{I}') = \emptyset$ for all $a \in \mathcal{N}_I$, it follows that $\mathcal{I}$ or $\mathcal{I}'$ is not coherent or that $\mathcal{I} = \mathcal{I}'$ holds. Since, by assumption, $\mathcal{I}$ and $\mathcal{I}'$ are coherent, $\mathcal{I} = \mathcal{I}'$ must hold. $\square$

The notions of conflict-free and refining $\mathcal{DL}$-interpretations can be extended to equivalence classes of coherent $\mathcal{DL}$-interpretations in the following sense.

Definition 7.3.24: Conflict-free and Refining Equivalence Classes

Let $\mathcal{R}$ be a knowledge base in $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$, let $\mathfrak{I}^{\sim}_{\mathcal{R}}$ be the set of the equivalence classes of $\mathcal{R}$-equivalent coherent $\mathcal{DL}$-interpretations (cf. Definition 7.3.17), and let $\Sigma^{f'}_{\mathcal{LA}}$ be a finite set of constraint representatives. For $[\mathcal{I}]_{\mathcal{R}} \in \mathfrak{I}^{\sim}_{\mathcal{R}}$ and $\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})$, we say that

- $[\mathcal{I}]_{\mathcal{R}}$ is *conflict-free* with $\mathcal{I}'$ iff there is $\mathcal{I}'' \in [\mathcal{I}]_{\mathcal{R}}$ such that $\mathcal{I}''$ is conflict-free with $\mathcal{I}'$.
- $[\mathcal{I}]_{\mathcal{R}}$ *refines* $\mathcal{I}'$ iff for all $\mathcal{I}'' \in [\mathcal{I}]_{\mathcal{R}}$ it holds that $\mathcal{I}''$ refines $\mathcal{I}'$.

We denote with $\mathsf{cf}_{\mathcal{R}}(\mathcal{I}')$ the set of equivalence classes from $\mathfrak{I}^{\sim}_{\mathcal{R}}$ which are conflict-free with $\mathcal{I}'$ and with $\mathsf{rf}_{\mathcal{R}}(\mathcal{I}')$ the set of equivalence classes which refine $\mathcal{I}'$.

Informally speaking, if $[\mathcal{I}]_{\mathcal{R}}$ refines $\mathcal{I}'$, then $\mathcal{I}'$ holds in every $\mathcal{DL}$-interpretation from $[\mathcal{I}]_{\mathcal{R}}$, and if $[\mathcal{I}]_{\mathcal{R}}$ is conflict-free with $\mathcal{I}'$, then $\mathcal{I}'$ cannot be proven wrong based on the $\mathcal{DL}$-interpretations in $[\mathcal{I}]_{\mathcal{R}}$. Hence, $\mathcal{I}'$ should hold at least with the probability of its refining equivalence classes and at most with the probability of its conflict-free equivalence classes. Before we formalize this, we prove that $\mathsf{rf}_{\mathcal{R}}(\mathcal{I}') \subseteq \mathsf{cf}_{\mathcal{R}}(\mathcal{I}')$ holds such that the equivalence classes in $\mathsf{rf}_{\mathcal{R}}(\mathcal{I}')$ indeed yield a lower probability for $\mathcal{I}$ than the equivalence classes in $\mathsf{cf}_{\mathcal{R}}(\mathcal{I}')$.

Proposition 7.3.25: Relationship Between Conflict-free and Refining

Let $\mathcal{R}$ be a knowledge base in $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$, let $\mathfrak{I}^{\sim}_{\mathcal{R}}$ be the set of the equivalence classes of $\mathcal{R}$-equivalent coherent $\mathcal{DL}$-interpretations (cf. Definition 7.3.17), and let $\Sigma^{f'}_{\mathcal{LA}}$ be a finite set of constraint representatives. For $\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})$, it holds that $\mathsf{rf}_{\mathcal{R}}(\mathcal{I}') \subseteq \mathsf{cf}_{\mathcal{R}}(\mathcal{I}')$.

Proof. Let $[\mathcal{I}]_{\mathcal{R}} \in \mathsf{rf}_{\mathcal{R}}(\mathcal{I}')$. Then, there is $\mathcal{I}'' \in [\mathcal{I}]_{\mathcal{R}}$ such that $\mathcal{I}''$ refines $\mathcal{I}'$. That is, $\mathcal{I}''$ and $\mathcal{I}'$ are logically identical and $\mathsf{vconf}_a(\mathcal{I}'') \subseteq \mathsf{vconf}_a(\mathcal{I}')$ for $a \in \mathcal{N}_I$ holds from which $\mathsf{vconf}_a(\mathcal{I}'') = \mathsf{vconf}_a(\mathcal{I}'') \cap \mathsf{vconf}_a(\mathcal{I}')$ for $a \in \mathcal{N}_I$ follows. Because for coherent $\mathcal{DL}$-interpretations it is $\mathsf{vconf}_a(\mathcal{I}') \neq \emptyset$ for $a \in \mathcal{N}_I$, it follows that $\mathsf{vconf}_a(\mathcal{I}'') \cap \mathsf{vconf}_a(\mathcal{I}') \neq \emptyset$ for $a \in \mathcal{N}_I$ as well. Hence, $\mathcal{I}''$ is conflict-free with $\mathcal{I}'$ which proves that $[\mathcal{I}]_{\mathcal{R}}$ is conflict-free with $\mathcal{I}'$ from which the validity of the proposition follows. ☐

Now, we are prepared to infer probability intervals of coherent $\mathcal{DL}$-interpretations from the aggregated MaxEnt-model.

Definition 7.3.26: Probability Intervals of $\mathcal{DL}$-interpretations

Let $\mathcal{R}$ be a consistent $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-knowledge base with aggregated MaxEnt-model $\widetilde{\mathcal{P}}^{\mathsf{ME}}_{\mathcal{R}}$, let $\Sigma^{f'}_{\mathcal{LA}}$ be a finite set of constraint representatives, and let $\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})$ be a coherent $\mathcal{DL}$-interpretation. Then, we define the *MaxEnt probability interval* $[l^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}'), u^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}')] \subseteq [0,1]$ of $\mathcal{I}'$ by

$$l^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}') = \sum_{[\mathcal{I}]_{\mathcal{R}} \in \mathsf{rf}_{\mathcal{R}}(\mathcal{I}')} \widetilde{\mathcal{P}}^{\mathsf{ME}}_{\mathcal{R}}([\mathcal{I}]_{\mathcal{R}}) \quad \text{and} \quad u^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}') = \sum_{[\mathcal{I}]_{\mathcal{R}} \in \mathsf{cf}_{\mathcal{R}}(\mathcal{I}')} \widetilde{\mathcal{P}}^{\mathsf{ME}}_{\mathcal{R}}([\mathcal{I}]_{\mathcal{R}}).$$

We write $\mathcal{R} \mid\!\!\sim^{\sim}_{\mathsf{ME}} \mathcal{I}'[l^{\mathsf{ME}}_{\mathcal{R}}(I'), u^{\mathsf{ME}}_{\mathcal{R}}(I')]$ in order to indicate that the MaxEnt probability interval of $\mathcal{I}'$ inferred from $\mathcal{R}$ is $[l^{\mathsf{ME}}_{\mathcal{R}}(I'), u^{\mathsf{ME}}_{\mathcal{R}}(I')]$.

The MaxEnt probability interval $[l^{\mathsf{ME}}_{\mathcal{R}}(I'), u^{\mathsf{ME}}_{\mathcal{R}}(I')]$ of I' states that it is reasonable to assume that the probability of I' is somewhere between $l^{\mathsf{ME}}_{\mathcal{R}}(I')$ and $u^{\mathsf{ME}}_{\mathcal{R}}(I')$ given the prior beliefs in $\mathcal{R}$.

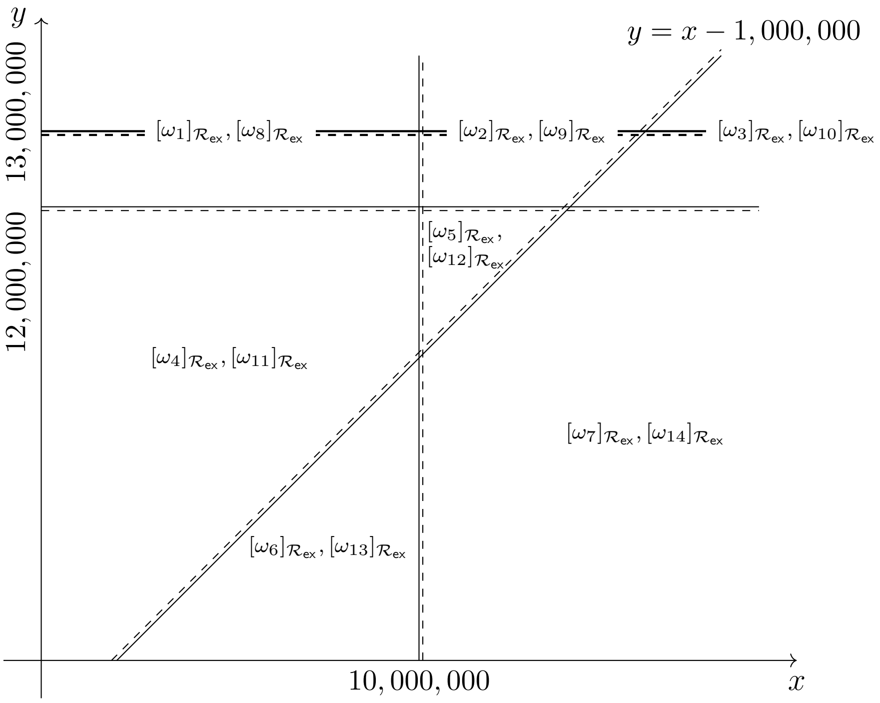

Figure 7.2: Areas of the admissible variable configurations of the equivalence classes $[\omega_1]_{\mathcal{R}_{\mathsf{ex}}}, \ldots, [\omega_{14}]_{\mathcal{R}_{\mathsf{ex}}}$ and the additional constraint m (thicker) from Example 7.3.27. Dashed lines indicate that the limit is still within the area.

Example 7.3.27

We consider the aggregated MaxEnt model of the belief base $\mathcal{R}_{\mathsf{ex}}$ from Example 7.3.13 as shown in Table 7.4. Suppose that the shareholder wants to investigate how likely a dividend payout is if the expenses do not exceed €13,000,000. For this, she formulates the constraint

$$m\colon \qquad\qquad y \leq 13,000,000.$$

"The expenses are at <u>m</u>ost €13,000,000."

See Figure 7.2 for a visualization of the admissible variable configurations of the equivalence classes over which the aggregated MaxEnt model is built as well as the constraint m. In a first step, the shareholder is interested in the probability of the coherent $\mathcal{DL}$-interpretation $da_m \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})$,[a] where $\Sigma^{f'}_{\mathcal{LA}} = \{a_m\}$. We obtain

$$\mathsf{rf}_{\mathcal{R}_{\mathsf{ex}}}(da_m) = \{[\omega_4]_{\mathcal{R}_{\mathsf{ex}}},\ [\omega_5]_{\mathcal{R}_{\mathsf{ex}}},\ [\omega_6]_{\mathcal{R}_{\mathsf{ex}}},\ [\omega_7]_{\mathcal{R}_{\mathsf{ex}}}\}$$

because $e\colon y \leq 12,000,000$ refines m (every solution of e is also a solution of m) and $[\omega_4]_{\mathcal{R}_{\mathsf{ex}}}, [\omega_5]_{\mathcal{R}_{\mathsf{ex}}}, [\omega_6]_{\mathcal{R}_{\mathsf{ex}}}$, and $[\omega_7]_{\mathcal{R}_{\mathsf{ex}}}$ satisfy a_e and d, while $[\omega_1]_{\mathcal{R}_{\mathsf{ex}}}, [\omega_2]_{\mathcal{R}_{\mathsf{ex}}}$, and $[\omega_3]_{\mathcal{R}_{\mathsf{ex}}}$ do not satisfy a_e, and all remaining equivalence classes do not satisfy d. Further,

$$\mathsf{cf}_{\mathcal{R}_{\mathsf{ex}}}(da_m) = \mathsf{rf}_{\mathcal{R}}(da_m) \cup \{[\omega_1]_{\mathcal{R}_{\mathsf{ex}}},\ [\omega_2]_{\mathcal{R}_{\mathsf{ex}}},\ [\omega_3]_{\mathcal{R}_{\mathsf{ex}}}\}$$

holds, basically because $\widehat{e}\colon y > 12,000,000$ and m are not in conflict, i.e., $[\omega_1]_{\mathcal{R}_{\mathsf{ex}}}, [\omega_2]_{\mathcal{R}_{\mathsf{ex}}}$, and $[\omega_3]_{\mathcal{R}_{\mathsf{ex}}}$, which satisfy $\overline{a_e}$ and d, have admissible variable configurations which satisfy m. As a consequence, the probability of da_m should be between

$$l^{\mathsf{ME}}_{\mathcal{R}_{\mathsf{ex}}}(da_m) = \sum_{i \in \{4,\dots,7\}} \widetilde{\mathcal{P}}^{\mathsf{ME}}_{\mathcal{R}_{\mathsf{ex}}}([\omega_i]_{\mathcal{R}_{\mathsf{ex}}}) \approx 0.572$$

and

$$u^{\mathsf{ME}}_{\mathcal{R}_{\mathsf{ex}}}(da_m) = \sum_{i \in \{1,\dots,7\}} \widetilde{\mathcal{P}}^{\mathsf{ME}}_{\mathcal{R}_{\mathsf{ex}}}([\omega_i]_{\mathcal{R}_{\mathsf{ex}}}) \approx 0.635.$$

The exact probability is unknown because we do not know to what extent the probabilities of $[\omega_1]_{\mathcal{R}_{\mathsf{ex}}}, [\omega_2]_{\mathcal{R}_{\mathsf{ex}}}$, and $[\omega_3]_{\mathcal{R}_{\mathsf{ex}}}$ contribute to the probability of da_m.

---

[a] Note that we denote $\mathcal{DL}$-interpretations in the form of possible worlds here.

The MaxEnt probability intervals of $\mathcal{DL}$-interpretations can be used to draw inferences.

Definition 7.3.28: Aggregated Maximum Entropy Inference

Let $\mathcal{R}$ be a consistent $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-knowledge base with aggregated MaxEnt-model $\widetilde{\mathcal{P}}^{\mathsf{ME}}_{\mathcal{R}}$, and let $\Sigma^{f'}_{\mathcal{LA}}$ be a set of constraint representatives. Then,

- $\mathcal{R} \mathrel{\widetilde{\mid\!\sim}}_{\mathsf{ME}} C(a)[l,u]$ iff
$$l = \sum_{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})\colon a^{\mathcal{I}'} \in C^{\mathcal{I}'}} l^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}'), \qquad u = \sum_{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})\colon a^{\mathcal{I}'} \in C^{\mathcal{I}'}} u^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}'),$$
- $\mathcal{R} \mathrel{\widetilde{\mid\!\sim}}_{\mathsf{ME}} r(a,b)[l,u]$ iff
$$l = \sum_{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})\colon (a^{\mathcal{I}'}, b^{\mathcal{I}'}) \in r^{\mathcal{I}'}} l^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}'), \qquad u = \sum_{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})\colon (a^{\mathcal{I}'}, b^{\mathcal{I}'}) \in r^{\mathcal{I}'}} u^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}'),$$
- $\mathcal{R} \mathrel{\widetilde{\mid\!\sim}}_{\mathsf{ME}} C \sqsubseteq D[l,u]$ iff
$$l = \sum_{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})\colon C^{\mathcal{I}'} \subseteq D^{\mathcal{I}'}} l^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}'), \qquad u = \sum_{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})\colon C^{\mathcal{I}'} \subseteq D^{\mathcal{I}'}} u^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}'),$$

- $\mathcal{R} \mathrel{|\!\!\!\sim}^{\sim}_{\mathsf{ME}} (D|C)[l,u]$ iff

$$\sum_{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})} |(C \sqcap D)^{\mathcal{I}'}| \cdot l^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}') + \sum_{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})} |(C \sqcap \neg D)^{\mathcal{I}'}| \cdot u^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}') > 0,$$

and

$$l = \frac{\displaystyle\sum_{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})} |(C \sqcap D)^{\mathcal{I}'}| \cdot l^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}')}{\displaystyle\sum_{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})} |(C \sqcap D)^{\mathcal{I}'}| \cdot l^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}') + \sum_{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})} |(C \sqcap \neg D)^{\mathcal{I}'}| \cdot u^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}')},$$

$$u = \frac{\displaystyle\sum_{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})} |(C \sqcap D)^{\mathcal{I}'}| \cdot u^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}')}{\displaystyle\sum_{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})} |(C \sqcap D)^{\mathcal{I}'}| \cdot u^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}') + \sum_{\mathcal{I}' \in \mathfrak{I}_c(\Sigma^{f'}_{\mathcal{LA}})} |(C \sqcap \neg D)^{\mathcal{I}'}| \cdot l^{\mathsf{ME}}_{\mathcal{R}}(\mathcal{I}')},$$

where $a, b \in \mathcal{N}_I$, $r \in \mathcal{N}_R$, and C, D are $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-concepts.

Note that the inferred probability intervals $[l,u]$ in Definition 7.3.28 are indeed subsets of $[0,1]$ by construction. For the lower probability bounds of axioms ($C(a)$, $r(a,b)$, and $C \sqsubseteq D$), the lower probability bounds of the $\mathcal{DL}$-interpretations are considered, and for the upper probability bounds of the axioms the upper probability bounds of the $\mathcal{DL}$-interpretations. In case of $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$-conditionals, we guarantee lower and upper probability bounds by a proper selection of the lower and upper probability bounds of the $\mathcal{DL}$-interpretations according to their evaluation of the conditional. In the absence of constraint representatives $l = u$ holds, and the definition of the inference relation $\mathrel{|\!\!\!\sim}^{\sim}_{\mathsf{ME}}$ reduces to the standard definition of maximum entropy inference.

Example 7.3.29

We continue Example 7.3.27 and consider the aggregated MaxEnt model of the belief base $\mathcal{R}_{\mathsf{ex}}$ from Example 7.3.13 as shown in Table 7.4 and the constraint

$$m: \qquad y \leq 13{,}000{,}000.$$

"The expenses are at <u>m</u>ost €13, 000, 000."

We (approximately) infer (cf. Example 7.3.27)

$$\mathcal{R}_{\mathsf{ex}} \mathrel{|\!\!\!\sim}^{\sim}_{\mathsf{ME}} (da_m)[0.572, 0.635]$$

> "Based on $\mathcal{R}_{\mathsf{ex}}$, it holds with a MaxEnt probability between 0.572 and 0.635 that the expenses do not exceed €13,000,000 and a dividend is paid."

Analogously, we have

$$\begin{aligned}\mathsf{rf}_{\mathcal{R}_{\mathsf{ex}}}(\bar{d}a_m) &= \{[\omega_{11}]_{\mathcal{R}_{\mathsf{ex}}},\ [\omega_{12}]_{\mathcal{R}_{\mathsf{ex}}},\ [\omega_{13}]_{\mathcal{R}_{\mathsf{ex}}},\ [\omega_{14}]_{\mathcal{R}_{\mathsf{ex}}}\},\\ \mathsf{cf}_{\mathcal{R}_{\mathsf{ex}}}(\bar{d}a_m) &= \mathsf{rf}_{\mathcal{R}_{\mathsf{ex}}}(da_m) \cup \{[\omega_8]_{\mathcal{R}_{\mathsf{ex}}},\ [\omega_9]_{\mathcal{R}_{\mathsf{ex}}},\ [\omega_{10}]_{\mathcal{R}_{\mathsf{ex}}}\},\end{aligned}$$

so that

$$l^{\mathsf{ME}}_{\mathcal{R}_{\mathsf{ex}}}(\bar{d}a_m) = \sum_{i\in\{11,\ldots,14\}} \widetilde{\mathcal{P}}^{\mathsf{ME}}_{\mathcal{R}_{\mathsf{ex}}}([\omega_i]_{\mathcal{R}_{\mathsf{ex}}}) \approx 0.328$$

and

$$u^{\mathsf{ME}}_{\mathcal{R}_{\mathsf{ex}}}(\bar{d}a_m) = \sum_{i\in\{8,\ldots,14\}} \widetilde{\mathcal{P}}^{\mathsf{ME}}_{\mathcal{R}_{\mathsf{ex}}}([\omega_i]_{\mathcal{R}_{\mathsf{ex}}}) \approx 0.364,$$

and, thus, $\mathcal{R}_{\mathsf{ex}} \mathrel{|\!\!\!\sim}^{\sim}_{\mathsf{ME}} (\bar{d}a_m)[0.328, 0.364]$. With

$$\frac{0.572}{0.572+0.364} = 0.611, \qquad \frac{0.635}{0.635+0.328} = 0.659,$$

we (approximatively) obtain

$$\mathcal{R}_{\mathsf{ex}} \mathrel{|\!\!\!\sim}^{\sim}_{\mathsf{ME}} (d|a_m)[0.611; 0.659].$$

That is, under the shareholder's assumption that the company's expenses stay less than €13,000,000, she should believe in the payout of the dividend with a probability of at least 0.611 and at most 0.659 according to her prior beliefs in $\mathcal{R}_{\mathsf{ex}}$ and the (aggregated) MaxEnt principle.

In this section we have made a proposal for integrating linear arithmetic constraints into the probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ which resulted in $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$. Linear arithmetic constraints specify variable configurations over infinite domains and, therefore, allow us to express beliefs formulated over infinite domains. However, the specification of variable configurations with linear arithmetic constraints is not unique. Thus, we had to abstract from the syntactic representation of variable configurations and consider equivalence classes. Based on this abstraction, we defined a novel maximum entropy inference relation $\mathrel{|\!\!\!\sim}^{\sim}_{\mathsf{ME}}$. A more in-depth analysis of this inference relation, e.g., in terms of inference properties, remains future work. Also future work is to investigate if inferences in $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$ can be drawn in a domain-lifted manner.

The integration of linear arithmetic constraints into Description Logics as proposed in this section is related to the concept of Description Logics with concrete domains [Baader and Hanschke, 1991; Baader and Bortoli, 2023, 2024]. Concrete domains have been introduced in Description Logics to enable reference to concrete objects, e.g., to numbers (like in our case). For instance, the constraint $m\colon y \leq 13,000,000$ ("the expenses are at most € 13,000,000") from Example 7.3.27 could similarly be expressed in $\mathcal{ALC}$ with concrete domains by the concept $\leq_{13{,}000{,}000}$ (expenses). The main difference between $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$ and $\mathcal{ALC}$ with concrete domains is that in the latter approach the concrete domain is included in the $\mathcal{DL}$-interpretations as a parameter. Therewith, the number of $\mathcal{DL}$-interpretations is infinite if the concrete domain is $\mathbb{Z}$ or $\mathbb{R}$, even if the (abstract) domain Δ is finite. This undermines our goal of keeping the number of interpretations finite. A more detailed elaboration of the connections between my approach on integrating linear arithmetic constraints into Description Logics and Description Logics with concrete domains remains for future work.

8 Conclusions

In this final chapter of the thesis, we recap our investigations on drawing probabilistic inferences under maximum entropy for Description Logics. In Section 8.1, we summarize the main results of this thesis. Afterwards, in Section 8.2, we discuss these results in the context of the research questions from Section 1.3 and point out possible research directions for future work.

8.1 Summary

In this thesis, we investigated the task of drawing inferences from relational probabilistic conditional knowledge bases at maximum entropy (MaxEnt) and transferred our findings to the novel probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$. In Chapter 1, we briefly motivated the topic (Section 1.1) and discussed the state of the art (Section 1.2). After that, we formulated four research questions (Section 1.3) addressing the problems of systematically and efficiently calculating conditional structures, maximum entropy probabilities, as well as inferences from these probabilities. The research questions mention the problem of huge domain sizes in relational settings which can make reasoning intractable quite fast. They further ask how to combine probabilistic Description Logics with the principle of maximum entropy and address the problem of infinite domain sizes in the context of relational maximum entropy reasoning.

In Chapter 2, we settled the logical foundations of our investigations. We recalled the basics of propositional logics, relational logics, and Description Logics. We also compared the expressivity of these logics, in particular with respect to their application to knowledge representation and reasoning. As a common limitation, we noted that with none of these classical logics it is possible to formalize uncertain knowledge adequately. Therefore, we extended relational logics by a conditional operator in Chapter 3 and thereafter also by probabilities (Definition 3.3.1). Still in Chapter 3, we discussed with the aggregating semantics (Definition 3.4.1) an elaborate semantics of relational probabilistic conditionals and had a closer look at conditional structures (Definition 3.2.1) and feature functions (Definition 3.4.3) which are important constituents of the aggregating semantics. A first contribution in this chapter was the extension of conditional structures to sets of possible worlds (Definition 3.2.3).

Chapter 4 was dedicated to relational maximum entropy reasoning. We formalized the notions of probabilistic knowledge bases (Definition 4.1.1) and the probabilistic inductive inference task (Definition 4.1.16). We argued the need of a distinct probabilistic model of a knowledge base for drawing inferences, and motivated why the principle of maximum entropy provides a particularly good choice for this model selection task. After introducing the maximum entropy model for consistent knowledge bases as being the solution of a nonlinear optimization problem (Definition 4.2.1), we discussed the dual maximum entropy optimization problem in Section 4.3 (cf. Definition 4.3.1). The dual optimization provides a product representation of the maximum entropy model for which the feature functions of the knowledge base constitute a sufficient statistics (Proposition 4.3.2). We investigated what we call the maximum entropy equation system (Definition 4.3.7) and demonstrated the power of the product representation of the maximum entropy model when we showed that maximum entropy reasoning with respect to conditionals that are built of Boolean combinations of unary predicates can be completely reduced to propositional maximum entropy reasoning (Proposition 4.3.11, Proposition 4.3.13, and Proposition 4.3.12).

In Chapter 5, we conceptualized the task of lifted inference at maximum entropy (Section 5.1). The basic idea of lifted inference is to draw inferences in time at most polynomially in the domain size. An essential step towards lifted inference at maximum entropy is to find compact representations of the conditional structures which determine the maximum entropy model, and to develop problem-adapted methods for computing—more precisely, approximating—the maximum entropy model based on these compact representations. With condensed iterative scaling (Section 5.2) we proposed an algorithm for this task.

In Chapter 6, we presented typed model counting, the centerpiece of this thesis. Typed model counting is a variant of (algebraic) first-order model counting (Section 6.1) which allows us to incorporate techniques from first-order model counting and knowledge compilation into our conditional setting. We discussed the connection of typed model counting to algebraic model counting in Section 6.3 and used it to compute compact representations of the input of condensed iterative scaling in Section 6.4.

In Chapter 7, we eventually applied the principle of maximum entropy to Description Logics and developed with $\mathcal{ALC}^{\mathsf{ME}}$ a probabilistic extension of the prototypical Description Logic $\mathcal{ALC}$ (Section 7.1). We demonstrated how lifted inferences can be drawn from $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases in Section 7.2. The probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ is limited to fixed finite domains. In order to deal with infinite domains, which turned out to be a tough problem in the context of maximum entropy reasoning, we proposed an extension of $\mathcal{ALC}^{\mathsf{ME}}$ by linear arithmetic constraints in Section 7.3. The linear arithmetic constraints are formulated over infinite domains, both over the integers and reals, and their satisfiability is evaluated modulo theory. Therewith, the probability space remains finite.

In total, we developed with typed model counting and condensed iterative scaling, as well as $\mathcal{ALC}^{\mathsf{ME}}$, a firm basis for further investigations on relational maximum entropy reasoning. We discuss possible directions for future work on relational maximum entropy reasoning in the next section.

8.2 Discussion and Future Work

The main contribution of this thesis is the development of typed model counting (Section 6.2) and condensed iterative scaling (Section 5.2). With these approaches we provide a general framework for systematic computations of conditional structures and maximum entropy probabilities with respect to relational probabilistic knowledge bases. Therewith, we contribute to answering the research questions **Q1** and **Q2** (Section 1.3). Further, with the probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ we propose a way of integrating maximum entropy reasoning in Description Logics with fixed finite domains (cf. research question **Q3** and Section 7.1). Eventually, we address research question **Q4** by our approach on integrating infinite domains into relational maximum entropy reasoning via satisfiability modulo theory (Section 7.3).

Nevertheless, many matters of detail remain open and my research on probabilistic inferences under maximum entropy for Description Logics can be extended in various directions. To close this thesis, we broach some topics related to relational maximum entropy reasoning which I find especially important, and we point to possible future work.

Domain-Lifted Relational Maximum Entropy Reasoning: We have seen that there are types of relational probabilistic knowledge bases for which domain-lifted reasoning at maximum entropy is possible (cf. Proposition 4.3.13). For a more general investigation of domain-lifted reasoning at maximum entropy the question seems essential whether the objective function of the dual maximum entropy optimization problem (Definition 4.3.1) has a compact, "domain-lifted" representation, i.e., a representation that can be evaluated in time polynomial in the domain size. With typed model counting we proposed a framework to compute such compact representations. Further, we proposed sophisticated counting strategies (Section 6.4) and provided typical examples in the appendix (Section A.2). However, it remains future work to identify larger classes of knowledge bases for which compact representations exist, more precisely, which can be compiled into structured sentences for which typed model counting is domain-liftable.

It is known that in the two-variable fragment of first-order logic the weighted first-order model counting task can be solved in time polynomial in the number of domain elements [Van den Broeck et al., 2014; Kuzelka, 2021]. However, when more than two variables are allowed, one quickly runs into unresolved problems. For instance, it has been shown that a formula for the number of transitive relations on an arbitrary set cannot be a polynomial [Mala, 2022; Pfeiffer, 2004].[1] But, to the best of my knowledge, it is still not known whether a (non-polynomial) formula with an explicit representation exists at all. Hence, for further investigations of typed model counting it seems sensible to have a look at the two variable fragment of structured sentences first.

Probabilistic Description Logics and Maximum Entropy: With $\mathcal{ALC}^{\mathsf{ME}}$ (Section 7.1) we developed a probabilistic Description Logic with a maximum entropy-based semantics and showed that domain-lifted maximum entropy reasoning in $\mathcal{ALC}^{\mathsf{ME}}$ is conceivable (Section 7.2). Therewith, we addressed our research question **Q3** (Section 1.3). Of course, it would be interesting to consider Description Logics alternative to $\mathcal{ALC}$ as a background language as well. For example, it would be worth investigating how probabilistic Description Logics perform at maximum entropy when specific role constructors are allowed (cf. Section 2.4).

Implementation: In order to utilize our approaches and test whether they are capable of handling real world problems, it is necessary to implement them. In particular, implementing typed model counting is an important task for future work. Existing implementations of relational maximum entropy reasoning are based on the grounding semantics of open conditionals (KReator[2] [Thimm et al., 2010; Beierle et al., 2010; Finthammer and Thimm, 2012; Beierle et al., 2015]; cf. Definition 3.4.12 for a definition of the grounding semantics) or, in case of the aggregating semantics, involve at least one iteration over the set of possible worlds [Finthammer, 2017, 2012], hence, are not domain-lifted. Thus, we can expect a drastic speed up in relational maximum entropy reasoning, at least for some important classes of knowledge bases, when making our approaches available. For a justification of this claim, please see the empirical results of our implementation of iGIS, a predecessor of typed model counting and condensed iterative scaling, in [Wilhelm et al., 2018], and also the catalog

---

[1] A relation S on a non-empty set $\mathcal{S}$ is transitive if and only if for all $X, Y, Z \in \mathcal{S}$ it holds that $(X, Y) \in S$ and $(Y, Z) \in S$ implies $(X, Z) \in S$.

[2] `https://kreator-ide.sourceforge.net/` (July 9, 2024)

of knowledge bases in Section A.2 of the appendix of this thesis.

A big challenge when implementing typed model counting is the compilation of structured sentences into $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal form because, in order to perform this compilation efficiently, the implementation has to involve many different compilation strategies (cf. Section 6.4). The task of implementing the computation of $\mathsf{sd\text{-}DNNF}^{\mathcal{S}}$-normal forms is highly related to the implementation of efficient knowledge compilation strategies from first-order model counting. For this task, implementations already exist and are under constant development (cf. the software tools FastWFOMC[3] [van Bremen and Kuzelka, 2021], ForcLift[4] [Van den Broeck et al., 2011], Crane [Dilkas and Belle, 2023], LRC [Kazemi and Poole, 2016], and Alchemy[5] [Gogate and Domingos, 2016]). However, since we make use of structured sentences instead of sentences here, we cannot simply utilize the implementations of first-order model counting but have to reproduce their techniques within our setting. Compared with this, implementations of algebraic model counting could be directly used for typed model counting (cf. Section 6.3). However, to the best of our knowledge, (efficient) implementations of algebraic model counting do not exist yet.

Complexity Analysis of Relational Maximum Entropy: Also future work is the complexity analysis of relational maximum entropy reasoning. Of course, this task is closely linked to the question under what circumstances relational maximum entropy reasoning is domain-liftable. However, we can also ask for the combined complexity measured in both the domain size and the size of the input knowledge base as we have done in the complexity analysis of the consistency of $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases in [Baader et al., 2019].

For the complexity analysis, the considered problem has to be specified clearly. In this thesis, we focused on computing the maximum entropy vector (cf. Definition 4.3.1) instead of the maximum entropy distribution itself because the size of the domain of the maximum entropy distribution is exponential in the size of the domain of the problem, while the maximum entropy vector has constant size measured in the domain size. Under specific circumstances, there may be a polynomial-time approximation scheme for computing the maximum entropy vector (cf. [Wilhelm and Kern-Isberner, 2019; Mintz and Aswani, 2017]). However, in order to be able to correctly assess this result, one still needs to know how the approximation error for the maximum entropy vector affects the corresponding approximation of the MaxEnt probabilities. Investi-

---

[3] `https://github.com/jan-toth/FastWFOMC.jl` (July 9, 2024)

[4] `https://github.com/UCLA-StarAI/Forclift` (July 9, 2024)

[5] `https://github.com/damienstanton/alchemy-lite` (July 9, 2024)

gating this connection proved to be difficult because the errors reproduce differently depending on the feature function values of the single possible worlds.

Relational Maximum Entropy and Belief Change: In this thesis, we investigated the reasoning task of drawing probabilistic inferences. In addition to that, we had a look into consistency issues of $\mathcal{ALC}^{\mathsf{ME}}$-knowledge bases in [Baader et al., 2019], too. Of course, there are much more probabilistic reasoning tasks, and it would be interesting whether techniques presented here carry over to these tasks. For example, an ample research field for future work on (efficient) relational maximum entropy reasoning is belief change, particularly belief revision. In belief revision, a belief state, here the maximum entropy distribution, has to be adapted to new insights resp. beliefs. Please see [Kern-Isberner, 2001a,b, 2004] for a discussion of belief revision in the context of maximum entropy and also [Jeffrey, 1965; Pearl, 1988; Gärdenfors and Rott, 1995; Katsuno and Mendelzon, 1991; Darwiche and Pearl, 1997] for more fundamental work on belief change.

Relational Maximum Entropy and Infinite Domains: In Section 7.3, we have argued why it is difficult to consider (countably) infinite domains when applying the principle of maximum entropy (cf. research question **Q4** in Section 1.3). As a way out, we have developed an approach which indirectly incorporates statements about infinite domains into maximum entropy reasoning via satisfiability modulo theory (Section 7.3). An alternative solution would be to adapt the notion of maximum entropy to better fit to infinite domains. For example, in [Landes, 2021, 2023], a comparative notion of maximal entropy is introduced which can cope with countably infinite domains but has the disadvantage that maximal entropy distributions are not unique. Further, no constructive methods for computing maximal entropy distributions under the comparative notion of maximal entropy are known. Nevertheless, it would be interesting future work to combine the approach from [Landes, 2021, 2023] with our probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$. Investigations on uncountably infinite domains in the context of maximum entropy reasoning would also be interesting.

Further, the probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}_{\mathcal{LA}}$ seems to be closely related to Description Logics with concrete domains [Baader and Hanschke, 1991; Baader and Bortoli, 2023, 2024]. Developing this connection in more detail is also future work.

Relational Maximum Entropy and Cognitive Aspects: In the propositional case maximum entropy reasoning is well-justified by some fundamental commonsense principles [Paris, 1998]. It would be worth investigating whether these principles, or adaptions thereof, are also satisfied by the principle of maximum entropy under the aggregating semantics in the relational case. This would particularly be a strong justification for the aggregating semantics.

Knowlede Discovery: A question that we did not address in this thesis at all is the question where the considered knowledge bases come from. We always assumed that knowledge bases are given, for example, as part of an expert system. In the motivation of this thesis (Section 1.1), we have claimed that knowledge-based systems, in principle, bring along required qualifications to overcome some shortcomings of data-driven approaches to AI. This, however, demands for the possibility to acquire high-level knowledge from large data sets. In [Fisseler et al., 2007], the algorithm CondorCKD is proposed which is capable of extracting probabilistic conditionals from data which may serve as a base for inductive reasoning via maximum entropy, but only in the propositional case. An important yet challenging research field for future work is to develop algorithms that extract relational probabilistic conditionals from data with a reasonable meaning in the context of the aggregating semantics and the principle of maximum entropy.

A Appendix

This appendix consists of three parts. First, we give some mathematical background information on the main concepts of the thesis (Section A.1). Second, we list basic knowledge bases involving conditionals built upon formulas that correspond to typical concept constructors in Description Logics. We translate these knowledge bases into structured sentences and compute their typed model counts in dependence of the domain size (Section A.2). And third, we give a complete list of the peer-reviewed publications I contributed to so far (Section A.3).

A.1 Mathematical Foundations

In this section, we recall the definitions of selected mathematical concepts which play a role in this thesis. The definitions are in the same order as the concepts appear in the thesis.

Definition A.1.1: Partition *(cf. [Halmos, 1960])*

Let S be a set. A family of sets $S_1, \ldots, S_m$ is a partition of S iff

- $\bigcup_{i=1}^{m} S_i = S$,
- $S_i \neq \emptyset$ for all $i = 1, \ldots, m$,
- $S_i \cap S_j = \emptyset$ for all $i, j \in \{1, \ldots, m\}$ with $i \neq j$.

Definition A.1.2: Abelian Group *(cf. [Lang, 2002])*

A set $\mathcal{G}$ equipped with a binary operation $\circ$ on $\mathcal{G}$ is an *Abelian group* iff $\circ$ satisfies the following conditions:

- $\forall a, b, c \in \mathcal{G} \colon a \circ (b \circ c) = (a \circ b) \circ c,$ (associativity)
- $\forall a, b \in \mathcal{G} \colon a \circ b = b \circ a,$ (commutativity)

- $\exists e \in \mathcal{G}\ \forall a \in \mathcal{G}\colon a \circ e = a$, ($e$ is identity element)
- $\forall a \in \mathcal{G}\ \exists a^{-1} \in \mathcal{G}\colon a \circ a^{-1} = e$. (inverse element)

Definition A.1.3: Free Abelian Group *(cf. [Lang, 2002])*

An Abelian group $\mathcal{G}$ with binary operation $\circ$ is called *free* if there is a subset $\mathcal{G}_G \subseteq \mathcal{G}$ such that every element $a \in \mathcal{G}$ can be uniquely written as a $\mathbb{Z}$-linear combination of the elements in $\mathcal{G}_G$. If $\mathcal{G}$ is free, then we denote $\mathcal{G}$ with $\mathcal{G} = (\mathcal{G}_G, \circ, e)$ where e is the identity element in $\mathcal{G}$. We call $\mathcal{G}_G$ a *generating set* of $\mathcal{G}$.

Definition A.1.4: (Free) Abelian Semiring *(cf. [Golan, 1999])*

A set $\mathcal{S}$ equipped with two binary operations $\oplus$ and $\otimes$, called *addition* and *multiplication*, is a *semiring*, iff there are elements $e^{\oplus}$ and $e^{\otimes}$ in $\mathcal{S}$ such that for all $a, b, c \in \mathcal{S}$

- $(a \oplus b) \oplus c = a \oplus (b \oplus c)$, (associativity of $\oplus$)
- $e^{\oplus} \oplus a = a = a \oplus e^{\oplus}$, ($e^{\oplus}$ is identity element wrt. $\oplus$)
- $a \oplus b = b \oplus a$, (commutativity of $\oplus$)
- $(a \otimes b) \otimes c = a \otimes (b \otimes c)$, (associativity of $\otimes$)
- $e^{\otimes} \otimes a = a = a \otimes e^{\otimes}$, ($e^{\otimes}$ is identity element wrt. $\otimes$)
- $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$, (left distributivity of $\otimes$ over $\oplus$)
- $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$, (right distributivity of $\otimes$ over $\oplus$)
- $e^{\oplus} \otimes a = e^{\oplus} = a \otimes e^{\oplus}$. (annihilation of $e^{\oplus}$)

If, in addition,

- $a \otimes b = b \otimes a$, (commutativity of $\otimes$)

then $\mathcal{S}$ is called *commutative* or *Abelian*. The semiring $\mathcal{S}$ is *free* if there is a subset $\mathcal{S}_{\mathcal{G}}$ of $\mathcal{S}$, called *generating set*, such that every element in $\mathcal{S}$ can be uniquely generated by $\mathcal{S}_{\mathcal{G}}$.

Definition A.1.5: Equivalence Relation *(cf. [Wilder, 1965])*

Let S be a non-empty set. A binary relation $\sim \subseteq S \times S$ is an *equivalence relation* on S iff

- $\forall a \in S\colon (a,a) \in \sim$, (reflexivity)
- $\forall a,b \in S\colon (a,b) \in \sim \Rightarrow (b,a) \in \sim$, (symmetry)
- $\forall a,b,c \in S\colon (a,b) \in \sim$ and $(b,c) \in \sim \Rightarrow (a,c) \in \sim$. (transitivity)

Definition A.1.6: Slater's Condition *(cf. [Boyd and Vandenberghe, 2009])*

Let $D \subseteq \mathbb{R}^n$, and let $f_i\colon D \to \mathbb{R}$ for $i = 1, \dots, m$ be real-valued functions on D. Then, the functions f_i, $i = 1, \dots, m$, satisfy *Slater's condition* if there exists a point x in the relative interior of D with $f_i(x) < 0$ for $i = 1, \dots, m$.

Definition A.1.7: Kullback-Leibler Divergence
(cf. [MacKay, 2003; Kullback, 1959])

Let $\mathcal{P}$ and $\mathcal{Q}$ be discrete probability distributions on the sample space $\mathcal{X}$. Then, the *Kullback-Leibler divergence* or *relative entropy* from $\mathcal{Q}$ to $\mathcal{P}$ is defined by

$$\mathcal{KL}(\mathcal{P}, \mathcal{Q}) = \sum_{x \in \mathcal{X}} \mathcal{P}(x) \cdot \log\left(\frac{\mathcal{P}(x)}{\mathcal{Q}(x)}\right).$$

Definition A.1.8: Inclusion-Exclusion Principle
(cf. [Roberts and Tesman, 2009])

The inclusion-exclusion principle has many formalizations. In its most common form it counts the number of elements in the union of finite sets as follows.

Let $S_1, \dots, S_n$ be finite sets. Then, the *inclusion-exclusion principle* is the identity

$$|\bigcup_{j=1}^{n} S_j| = \sum_{\emptyset \neq \mathcal{I} \subseteq \{1,\dots,n\}} (-1)^{|\mathcal{I}|+1} \cdot |\bigcap_{i \in \mathcal{I}} S_i|.$$

A.2 Catalog of Knowledge Bases

We present a catalog of 20 knowledge bases and use these to show how different typed model counting techniques work. An overview of the knowledge bases is given in Table A.1. For every knowledge base $\mathcal{R}$ in this catalog, we compile $\mathcal{R}$ into the structured sentence $\Psi_{\mathcal{R}}$ and compute the polynomials $\tau_{\mathcal{R}}$ and $\tau_{\mathcal{R}}^{r_i}$ for $r_i \in \mathcal{B}_{\mathcal{R}}$ which serve as the input of the condensed iterative scaling algorithm (cf. Section 6.4 for the theoretical details). After that, we recall some knowledge bases from the literature. We do not apply typed model counting to these knowledge bases in detail but give a hint which techniques could be used by mentioning similarities to the knowledge bases in Table A.1.

In order to keep expressions short, we use $\mathsf{s}(x)$ as a placeholder within structured sentences in order to indicate that the formula is not smooth at this point. Hereby, x is the number of the missing ground atoms. When counting the typed models of the structured sentence, $\mathsf{s}(x)$ means that one has to multiply the respective model count with 2^x. If not stated otherwise, then we treat the domain size $m = |\mathsf{Const}_\Sigma|$ as a parameter (note that we use m instead of k for the domain size in this catalog of knowledge bases).

In some cases, we apply compilation strategies to structured sentences which are not discussed in the thesis. We briefly explain these compilation strategies here:

Paired Grounding With paired grounding we avoid that ground instances of a binary predicate $R/2$ are counted twice. This can happen if the predicate occurs in a structured sentence with transposed variables, i.e., if both $R(X,Y)$ and $R(Y,X)$ are mentioned. Then, we separate the case where $X = Y$ holds, duplicate the remaining structured sentence but with changed roles of X and Y (e.g., $A(Y) \wedge R(Y,X)$ becomes $A(X) \wedge R(X,Y)$), and restrict Y to satisfy $Y < X$ (we implicitly assume a total order on Const_Σ in this case). See [Van den Broeck, 2013] for the theoretical background and $\mathcal{R}_{16}$ as an example.

Skolemization With skolemization we avoid to count models of existentially quantified structured sentences. The basic idea is that it is much easier, instead of counting the models of a sentence $\exists X.A(X)$ directly—all interpretations are models of this sentence except for the one in which $\overline{A(x)}$ for all $c \in \mathsf{Const}_\Sigma$ holds—, to count the only model of the negated sentence $\neg\exists X.A(X) \equiv \forall X.\overline{A(X)}$ and to subtract this one model from the total number of interpretations. With this strategy of "indirect counting," we obtain a model count of $2^m - 1$ when $m = |\mathsf{Const}_\Sigma|$. Within the framework of typed model counting, this strategy can be realized as follows.

Suppose that a structured sentence ψ_S mentions a subsentence of the form $\exists X.\phi(X,Y)$ where $\phi(X,Y)$ is a formula from $\mathcal{RL}(\Sigma)$ with the free variables X

$\mathcal{R} = (\mathcal{F}_\mathcal{R}, \mathcal{B}_\mathcal{R})$	Facts in $\mathcal{F}_\mathcal{R}$	Conditionals in $\mathcal{B}_\mathcal{R}$
$\mathcal{R}_1$		$r_1 = (B(X)\|A(X))[p_1],\ \ r_2 = (D(X)\|C(X))[p_2]$
$\mathcal{R}_2$		$r_1 = (B(X)\|A(X))[p_1],\ \ r_2 = (C(X)\|A(X))[p_2]$
$\mathcal{R}_3$		$r_1 = (B(X)\|A(X))[p_1],\ \ r_2 = (D(c)\|C(c))[p_2]$
$\mathcal{R}_4$		$r_1 = (B(X)\|A(X))[p_1],\ \ r_2 = (C(c)\|A(c))[p_2]$
$\mathcal{R}_5$	$C(c)$	$r_1 = (B(X)\|A(X))[p_1]$
$\mathcal{R}_6$	$A(c)$	$r_1 = (B(X)\|A(X))[p_1]$
$\mathcal{R}_7$	$\forall X.C(X)$	$r_1 = (B(X)\|A(X))[p_1]$
$\mathcal{R}_8$	$\forall X.A(X)$	$r_1 = (B(X)\|A(X))[p_1]$
$\mathcal{R}_9$	$\exists X.C(X)$	$r_1 = (B(X)\|A(X))[p_1]$
$\mathcal{R}_{10}$	$\exists X.A(X)$	$r_1 = (B(X)\|A(X))[p_1]$
$\mathcal{R}_{11}$		$r_1 = (R(X,Y)\|A(X))[p_1]$
$\mathcal{R}_{12}$		$r_1 = (S(X,Y)\|R(X,Y))[p_1]$
$\mathcal{R}_{13}$		$r_1 = (S(Y,X)\|R(X,Y))[p_1]$
$\mathcal{R}_{14}$		$r_1 = (B(X)\|A(X) \wedge \neg A(Y))[p_1]$
$\mathcal{R}_{15}$		$r_1 = (A(Y)\|A(X) \wedge R(X,Y))[p_1]$
$\mathcal{R}_{16}$		$r_1 = (R(Y,X)\|R(X,Y))[p_1]$
$\mathcal{R}_{17}$		$r_1 = (R(X,Z)\|R(X,Y) \wedge R(Y,Z))[p_1]$
$\mathcal{R}_{18}$		$r_1 = (\forall Y.R(X,Y) \Rightarrow A(Y)\|A(X))[p_1]$
$\mathcal{R}_{19}$		$r_1 = (\forall Y.R(X,Y) \Rightarrow B(Y)\|A(X))[p_1]$
$\mathcal{R}_{20}$		$r_1 = (\exists Y.R(X,Y) \wedge B(Y)\|A(X))[p_1]$

Table A.1: Overview of the benchmark knowledge bases in this catalog.

and Y. Then, we introduce two fresh predicates $S/1$ and $Z/1$, replace the expression $\exists X.\phi(X,Y)$ by $Z(Y)$, and append the structured sentence

$$\forall Y.\left(Z(Y)(S(Y)\vee(-1)\circ\overline{S(Y)}\wedge\forall X.\overline{\phi(X,Y)})\vee\overline{Z(Y)}S(Y)\wedge\forall X.\overline{\phi(X,Y)}\right)$$

with structure element -1 to $\psi_{\mathcal{S}}$. This idea is based on Van den Broeck et al. [2014]. See $\mathcal{R}_{18}$ as an example.

▶ Knowledge Base $\mathcal{R}_1$

We consider $\mathsf{Pred}_\Sigma=\{A/1,B/1,C/1,D/1\}$ and $\mathcal{R}_1=(\mathcal{F},\mathcal{B})$ with

- $\mathcal{F}=\emptyset$,
- $\mathcal{B}=\{r_1,r_2\}$ with $r_1=(B(X)|A(X))[p_1],\quad r_2=(D(X)|C(X))[p_2]$.

$$\begin{aligned}\Psi_{\mathcal{R}_1}\equiv_{\mathcal{S}}&\ \forall X.\left[y_1A(X)\left(x_1B(x)\vee\overline{B(X)}\right)\vee\overline{A(X)}\right]\\&\ \forall X.\left[y_2C(X)\left(x_2D(x)\vee\overline{D(X)}\right)\vee\overline{C(X)}\right]\\\equiv_{\mathcal{S}}&\ \forall X.\left[\left(y_1A(X)\left(x_1B(x)\vee\overline{B(X)}\right)\vee\mathsf{s}(1)\overline{A(X)}\right)\right.\\&\qquad\left.\left(y_2C(X)\left(x_2D(x)\vee\overline{D(X)}\right)\vee\mathsf{s}(1)\overline{C(X)}\right)\right]\end{aligned}$$

$$\tau_{\mathcal{R}_1}=((y_1(x_1+1)+2)\ (y_2(x_2+1)+2))^m$$

Let $c\in\mathsf{Const}_\Sigma$. ($\Psi_{\mathcal{R}_1}C(c)$ and $\Psi_{\mathcal{R}_1}C(c)D(c)$ can be calculated analogously.)

$$\begin{aligned}\Psi_{\mathcal{R}_1}A(c)\equiv_{\mathcal{S}}&\ \forall X_{\neq c}.\left[\left(y_1A(X)\left(x_1B(x)\vee\overline{B(X)}\right)\vee\mathsf{s}(1)\overline{A(X)}\right)\right.\\&\qquad\left.\left(y_2C(X)\left(x_2D(x)\vee\overline{D(X)}\right)\vee\mathsf{s}(1)\overline{C(X)}\right)\right]\\&\ y_1A(c)\left(x_1B(c)\vee\overline{B(c)}\right)\left(y_2C(c)\left(x_2D(c)\vee\overline{D(c)}\right)\vee\mathsf{s}(1)\overline{C(c)}\right)\\\Psi_{\mathcal{R}_1}A(c)B(c)\equiv_{\mathcal{S}}&\ \forall X_{\neq c}.\left[\left(y_1A(X)\left(x_1B(x)\vee\overline{B(X)}\right)\vee\mathsf{s}(1)\overline{A(X)}\right)\right.\\&\qquad\left.\left(y_2C(X)\left(x_2D(x)\vee\overline{D(X)}\right)\vee\mathsf{s}(1)\overline{C(X)}\right)\right]\\&\ y_1A(c)\left(x_1B(c)\right)\left(y_2C(c)\left(x_2D(c)\vee\overline{D(c)}\right)\vee\mathsf{s}(1)\overline{C(c)}\right)\end{aligned}$$

$$\begin{aligned}\tau_{\mathcal{R}_1}^{r_1}&=m\ ((y_1(x_1+1)+2)\ (y_2(x_2+1)+2))^{m-1}\ (y_2(x_2+1)+2)\ (y_1x_1+zy_1(x_1+1))\\\tau_{\mathcal{R}_1}^{r_2}&=m\ ((y_1(x_1+1)+2)\ (y_2(x_2+1)+2))^{m-1}\ (y_1(x_1+1)+2)\ (y_2x_2+zy_2(x_2+1))\end{aligned}$$

▶ Knowledge Base $\mathcal{R}_2$

We consider $\mathsf{Pred}_\Sigma = \{A/1, B/1, C/1\}$ and $\mathcal{R}_2 = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1, r_2\}$ with $r_1 = (B(X)|A(X))[p_1], \quad r_2 = (C(X)|A(X))[p_2]$.

$$\Psi_{\mathcal{R}_2} \equiv_{\mathcal{S}} \forall X. \left[y_1 y_2 A(X) \left(x_1 B(X) \vee \overline{B(X)} \right) \left(x_2 C(X) \vee \overline{C(X)} \right) \vee \mathsf{s}(2)\overline{A(X)} \right]$$

$$\tau_{\mathcal{R}_2} = (y_1(x_1+1)y_2(x_2+1)+4)^m$$

Let $c \in \mathsf{Const}_\Sigma$. ($\Psi_{\mathcal{R}_2} A(c)C(c)$ can be calculated analogously to $\Psi_{\mathcal{R}_2} A(c)B(c)$.)

$$\begin{aligned}
\Psi_{\mathcal{R}_2} A(c) \equiv_{\mathcal{S}} {} & \forall X_{\neq c}. \left[y_1 y_2 A(X) \left(x_1 B(X) \vee \overline{B(X)} \right) \left(x_2 C(X) \vee \overline{C(X)} \right) \vee \mathsf{s}(2)\overline{A(X)} \right] \\
& y_1 y_2 A(c) \left(x_1 B(c) \vee \overline{B(c)} \right) \left(x_2 C(c) \vee \overline{C(c)} \right) \\
\Psi_{\mathcal{R}_2} A(c)B(c) \equiv_{\mathcal{S}} {} & \forall X_{\neq c}. \left[y_1 y_2 A(X) \left(x_1 B(X) \vee \overline{B(X)} \right) \left(x_2 C(X) \vee \overline{C(X)} \right) \vee \mathsf{s}(2)\overline{A(X)} \right] \\
& y_1 y_2 A(c) \left(x_1 B(c) \right) \left(x_2 C(c) \vee \overline{C(c)} \right)
\end{aligned}$$

$$\begin{aligned}
\tau_{\mathcal{R}_2}^{r_1} &= m \; (y_1(x_1+1)y_2(x_2+1)+4)^{m-1} \; y_2(x_2+1) \; (y_1 x_1 + z y_1(x_1+1)) \\
\tau_{\mathcal{R}_2}^{r_2} &= m \; (y_1(x_1+1)y_2(x_2+1)+4)^{m-1} \; y_1(x_1+1) \; (y_2 x_2 + z y_2(x_2+1))
\end{aligned}$$

▶ Knowledge Base $\mathcal{R}_3$

We consider $\mathsf{Pred}_\Sigma = \{A/1, B/1, C/1, D/1\}$, $c \in \mathsf{Const}_\Sigma$, and $\mathcal{R}_3 = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1, r_2\}$ with $r_1 = (B(X)|A(X))[p_1], \quad r_2 = (D(c)|C(c))[p_2]$.

$$\begin{aligned}
\Psi_{\mathcal{R}_3} \equiv_{\mathcal{S}} {} & \mathsf{s}(2(m-1)) \; \forall X. \left[y_1 A(X) \left(x_1 B(X) \vee \overline{B(X)} \right) \vee \mathsf{s}(1)\overline{A(X)} \right] \\
& \left(y_2 C(c) \left(x_2 D(c) \vee \overline{D(c)} \right) \vee \mathsf{s}(1)\overline{C(c)} \right)
\end{aligned}$$

$$\tau_{\mathcal{R}_3} = 2^{2(m-1)}\ (y_1(x_1+1)+2)^m\ (y_2(x_2+1)+2)$$

Let $d \in \mathsf{Const}_\Sigma$ regardless of whether $d = c$ or not.

$$\Psi_{\mathcal{R}_3} A(d) \equiv_{\mathcal{S}} \mathsf{s}(2(m-1))\ \forall X_{\neq d}.\ \Big[y_1 A(X)\ \Big(x_1 B(X) \vee \overline{B(X)}\Big) \vee \mathsf{s}(1)\overline{A(X)}\Big]$$

$$y_1 A(d)\ \Big(x_1 B(d) \vee \overline{B(d)}\Big)\ \Big(y_2 C(c)\ \Big(x_2 D(c) \vee \overline{D(c)}\Big) \vee \mathsf{s}(1)\overline{C(c)}\Big)$$

$$\Psi_{\mathcal{R}_3} A(d)B(d) \equiv_{\mathcal{S}} \mathsf{s}(2(m-1))\ \forall X_{\neq d}.\ \Big[y_1 A(X)\ \Big(x_1 B(X) \vee \overline{B(X)}\Big) \vee \mathsf{s}(1)\overline{A(X)}\Big]$$

$$y_1 A(d)\Big(x_1 B(d)\Big)\ \Big(y_2 C(c)\ \Big(x_2 D(c) \vee \overline{D(c)}\Big) \vee \mathsf{s}(1)\overline{C(c)}\Big)$$

$$\Psi_{\mathcal{R}_3} C(c) \equiv_{\mathcal{S}} \mathsf{s}(2(m-1))\ \forall X.\ \Big[y_1 A(X)\ \Big(x_1 B(X) \vee \overline{B(X)}\Big) \vee \mathsf{s}(1)\overline{A(X)}\Big]$$

$$y_2 C(c)\ \Big(x_2 D(c) \vee \overline{D(c)}\Big)$$

$$\Psi_{\mathcal{R}_3} C(c)D(c) \equiv_{\mathcal{S}} \mathsf{s}(2(m-1))\ \forall X.\ \Big[y_1 A(X)\ \Big(x_1 B(X) \vee \overline{B(X)}\Big) \vee \mathsf{s}(1)\overline{A(X)}\Big]$$

$$y_2 C(c)\Big(x_2 D(c)\Big)$$

$$\begin{aligned} \tau_{\mathcal{R}_3}^{r_1} &= m\ 2^{2(m-1)}\ (y_1(x_1+1)+2)^{m-1}\ (y_2(x_2+1)+2)\ (y_1x_1 + zy_1(x_1+1)) \\ \tau_{\mathcal{R}_3}^{r_2} &= \quad 2^{2(m-1)}\ (y_1(x_1+1)+2)^{m} \qquad\qquad (y_2x_2 + zy_2(x_2+1)) \end{aligned}$$

▶ Knowledge Base $\mathcal{R}_4$

We consider $\mathsf{Pred}_\Sigma = \{A/1, B/1, C/1\}$, $c \in \mathsf{Const}_\Sigma$, and $\mathcal{R}_3 = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1, r_2\}$ with $r_1 = (B(X)|A(X))[p_1]$, $\quad r_2 = (C(c)|A(c))[p_2]$.

$m = 1$

$$\Psi_{\mathcal{R}_4} \equiv_{\mathcal{S}} y_1 y_2 A(c)\Big(x_1 B(c) \vee \overline{B(c)}\Big)\Big(x_2 C(c) \vee \overline{C(c)}\Big) \vee \mathsf{s}(2)\overline{A(c)}$$

$$\tau_{\mathcal{R}_4} = (y_1(x_1+1)y_2(x_2+1)+4)$$

$$\Psi_{\mathcal{R}_4} A(c) \equiv_{\mathcal{S}} y_1 y_2 A(c)\Big(x_1 B(c) \vee \overline{B(c)}\Big)\Big(x_2 C(c) \vee \overline{C(c)}\Big)$$

$$\begin{array}{c}
\Psi_{\mathcal{R}_4} A(c) B(c) \equiv_{\mathcal{S}} y_1 y_2 A(c) \Big(x_1 B(c) \Big) \Big(x_2 C(c) \vee \overline{C(c)} \Big) \\
\hline
\Psi_{\mathcal{R}_4} A(c) C(c) \equiv_{\mathcal{S}} y_1 y_2 A(c) \Big(x_1 B(c) \vee \overline{B(c)} \Big) \Big(x_2 C(c) \Big)
\end{array}$$

$$\boxed{\begin{aligned}
\tau^{r_1}_{\mathcal{R}_4} &= y_2(x_2+1)\ (y_1 x_1 + z y_1 (x_1+1)) \\
\tau^{r_2}_{\mathcal{R}_4} &= y_1(x_1+1)\ (y_2 x_2 + z y_2 (x_2+1))
\end{aligned}}$$

$m > 1$

$$\begin{gathered}
\Psi_{\mathcal{R}_4} \equiv_{\mathcal{S}} \forall X_{\neq c}.\ \Big[y_1 A(X)\ \Big(x_1 B(X) \vee \overline{B(X)} \Big) \vee \mathsf{s}(1) \overline{A(X)} \Big]\ \mathsf{s}(|\mathsf{Const}_\Sigma| - 1) \\
\Big(y_1 y_2 A(c) \Big(x_1 B(c) \vee \overline{B(c)} \Big) \Big(x_2 C(c) \vee \overline{C(c)} \Big) \vee \mathsf{s}(2) \overline{A(c)} \Big)
\end{gathered}$$

$$\boxed{\tau_{\mathcal{R}_4} = 2^{m-1}\ (y_1(x_1+1)+2)^{m-1}\ (y_1(x_1+1) y_2 (x_2+1) + 4)}$$

$$\begin{gathered}
\Psi_{\mathcal{R}_4} A(c) \equiv_{\mathcal{S}} \forall X_{\neq c}.\ \Big[y_1 A(X)\ \Big(x_1 B(X) \vee \overline{B(X)} \Big) \vee \mathsf{s}(1) \overline{A(X)} \Big]\ \mathsf{s}(|\mathsf{Const}_\Sigma| - 1) \\
\Big(y_1 y_2 A(c) \Big(x_1 B(c) \vee \overline{B(c)} \Big) \Big(x_2 C(c) \vee \overline{C(c)} \Big) \Big) \\
\Psi_{\mathcal{R}_4} A(c) B(c) \equiv_{\mathcal{S}} \forall X_{\neq c}.\ \Big[y_1 A(X)\ \Big(x_1 B(X) \vee \overline{B(X)} \Big) \vee \mathsf{s}(1) \overline{A(X)} \Big]\ \mathsf{s}(|\mathsf{Const}_\Sigma| - 1) \\
\Big(y_1 y_2 A(c) \Big(x_1 B(c) \Big) \Big(x_2 C(c) \vee \overline{C(c)} \Big) \Big)
\end{gathered}$$

Let $d \in \mathsf{Const}_\Sigma$ with $d \neq c$.

$$\begin{array}{l}
\Psi_{\mathcal{R}_4} A(d) \equiv_{\mathcal{S}} \forall X_{\neq c,d}.\ \Big[y_1 A(X)\ \Big(x_1 B(X) \vee \overline{B(X)} \Big) \vee \mathsf{s}(1) \overline{A(X)} \Big]\ \mathsf{s}(|\mathsf{Const}_\Sigma| - 1) \\
\qquad y_1 A(d) \Big(x_1 B(d) \vee \overline{B(d)} \Big) \Big(y_1 y_2 A(c) \Big(x_1 B(c) \vee \overline{B(c)} \Big) \Big(x_2 C(c) \vee \overline{C(c)} \Big) \vee \mathsf{s}(2) \overline{A(c)} \Big) \\
\Psi_{\mathcal{R}_4} A(d) B(d) \equiv_{\mathcal{S}} \forall X_{\neq c,d}.\ \Big[y_1 A(X)\ \Big(x_1 B(X) \vee \overline{B(X)} \Big) \vee \mathsf{s}(1) \overline{A(X)} \Big]\ \mathsf{s}(|\mathsf{Const}_\Sigma| - 1) \\
\qquad y_1 A(d) \Big(x_1 B(d) \Big) \Big(y_1 y_2 A(c) \Big(x_1 B(c) \vee \overline{B(c)} \Big) \Big(x_2 C(c) \vee \overline{C(c)} \Big) \vee \mathsf{s}(2) \overline{A(c)} \Big) \\
\hline
\Psi_{\mathcal{R}_4} A(c) C(c) \equiv_{\mathcal{S}} \forall X_{\neq c}.\ \Big[y_1 A(X)\ \Big(x_1 B(X) \vee \overline{B(X)} \Big) \vee \mathsf{s}(1) \overline{A(X)} \Big]\ \mathsf{s}(|\mathsf{Const}_\Sigma| - 1) \\
\qquad \Big(y_1 y_2 A(c) \Big(x_1 B(c) \vee \overline{B(c)} \Big) \Big(x_2 C(c) \Big) \Big)
\end{array}$$

$$\boxed{\begin{aligned}
\tau^{r_1}_{\mathcal{R}_4} &= 2^{m-1}\ (y_1(x_1+1)+2)^{m-2}\ (m\ (y_1(x_1+1) y_2(x_2+1)+4) + \\
&\quad (y_1(x_1+1)+2)\ y_2(x_2+1))\ (y_1 x_1 + z y_1 (x_1+1)) \\
\tau^{r_2}_{\mathcal{R}_4} &= 2^{m-1}\ (y_1(x_1+1)+2)^{m-1}\ y_1(x_1+1)\ (y_2 x_2 + z y_2 (x_2+1))
\end{aligned}}$$

▶ Knowledge Base $\mathcal{R}_5$

We consider $\mathsf{Pred}_\Sigma = \{A/1, B/1, C/1\}$, $c \in \mathsf{Const}_\Sigma$, and $\mathcal{R}_5 = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \{C(c)\}$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (B(X)|A(X))[p_1]$.

$$\Psi_{\mathcal{R}_5} \equiv_{\mathcal{S}} \mathsf{s}(m-1)\ C(c)\ \forall X.\ \Big[y_1 A(X)\ \Big(x_1 B(X) \vee \overline{B(X)}\Big) \vee \mathsf{s}(1)\overline{A(X)}\Big]$$

$$\tau_{\mathcal{R}_5} = 2^{m-1}\ (y_1(x_1+1)+2)^m$$

Let $d \in \mathsf{Const}_\Sigma$ regardless of whether $d = c$ or not.

$$\Psi_{\mathcal{R}_5} A(d) \equiv_{\mathcal{S}} \mathsf{s}(m-1)\ C(c)\ \forall X_{\neq d}.\ \Big[y_1 A(X)\ \Big(x_1 B(X) \vee \overline{B(X)}\Big) \vee \mathsf{s}(1)\overline{A(X)}\Big]$$
$$y_1 A(d)\Big(x_1 B(d) \vee \overline{B(d)}\Big)$$
$$\Psi_{\mathcal{R}_5} A(d)B(d) \equiv_{\mathcal{S}} \mathsf{s}(m-1)\ C(c)\ \forall X_{\neq d}.\ \Big[y_1 A(X)\ \Big(x_1 B(X) \vee \overline{B(X)}\Big) \vee \mathsf{s}(1)\overline{A(X)}\Big]$$
$$y_1 A(d)\Big(x_1 B(d)\Big)$$

$$\tau^{r_1}_{\mathcal{R}_5} = m\ 2^{m-1}\ (y_1(x_1+1)+2)^m\ (y_1 x_1 + z y_1(x_1+1))$$

▶ Knowledge Base $\mathcal{R}_6$

We consider $\mathsf{Pred}_\Sigma = \{A/1, B/1, C/1\}$, $c \in \mathsf{Const}_\Sigma$, and $\mathcal{R}_6 = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \{A(c)\}$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (B(X)|A(X))[p_1]$.

$m = 1$

$$\Psi_{\mathcal{R}_6} \equiv_{\mathcal{S}} y_1 A(c)\Big(x_1 B(c) \vee \overline{B(c)}\Big)$$

$$\tau_{\mathcal{R}_6} = y_1(x_1+1)$$

$$\Psi_{\mathcal{R}_6} A(c) \equiv_{\mathcal{S}} \Psi_{\mathcal{R}_6}$$
$$\Psi_{\mathcal{R}_6} A(c)B(c) \equiv_{\mathcal{S}} y_1 A(c)\Big(x_1 B(c)\Big)$$

$$\boxed{\tau^{r_1}_{\mathcal{R}_6} = (y_1 x_1 + z y_1(x_1+1))}$$

$m > 1$

$$\begin{aligned}\Psi_{\mathcal{R}_6} \equiv_{\mathcal{S}}\ & \forall X_{\neq c}. \Big[y_1 A(X)\Big(x_1 B(X) \vee \overline{B(X)}\Big) \vee \mathsf{s}(1)\overline{A(X)}\Big] \\ & y_1 A(c)\Big(x_1 B(c) \vee \overline{B(c)}\Big)\end{aligned}$$

$$\boxed{\tau_{\mathcal{R}_6} = (y_1(x_1+1)+2)^{m-1}\ y_1(x_1+1)}$$

$$\Psi_{\mathcal{R}_6} A(c) \equiv_{\mathcal{S}} \Psi_{\mathcal{R}_6}$$
$$\begin{aligned}\Psi_{\mathcal{R}_6} A(c)B(c) \equiv_{\mathcal{S}}\ & \forall X_{\neq c}. \Big[y_1 A(X)\Big(x_1 B(X) \vee \overline{B(X)}\Big) \vee \mathsf{s}(1)\overline{A(X)}\Big] \\ & y_1 A(c)\Big(x_1 B(c)\Big)\end{aligned}$$

Let $d \in \mathsf{Const}_\Sigma$ with $d \neq c$.

$$\begin{aligned}\Psi_{\mathcal{R}_6} A(d) \equiv_{\mathcal{S}}\ & \forall X_{\neq c,d}. \Big[y_1 A(X)\Big(x_1 B(X) \vee \overline{B(X)}\Big) \vee \mathsf{s}(1)\overline{A(X)}\Big] \\ & y_1 A(c)\Big(x_1 B(c) \vee \overline{B(c)}\Big) y_1 A(d)\Big(x_1 B(d) \vee \overline{B(d)}\Big) \\ \Psi_{\mathcal{R}_6} A(d)B(d) \equiv_{\mathcal{S}}\ & \forall X_{\neq c,d}. \Big[y_1 A(X)\Big(x_1 B(X) \vee \overline{B(X)}\Big) \vee \mathsf{s}(1)\overline{A(X)}\Big] \\ & y_1 A(c)\Big(x_1 B(c) \vee \overline{B(c)}\Big) y_1 A(d)\Big(x_1 B(d)\Big)\end{aligned}$$

$$\boxed{\begin{aligned}\tau^{r_1}_{\mathcal{R}_6} = & ((y_1(x_1+1)+2) + (m-1)\ y_1(x_1+1))\ (y_1(x_1+1)+2)^{m-2} \\ & (y_1 x_1 + z y_1(x_1+1))\end{aligned}}$$

▶ Knowledge Base $\mathcal{R}_7$

We consider $\mathsf{Pred}_\Sigma = \{A/1, B/1, C/1\}$, and $\mathcal{R}_7 = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \{\forall X.C(X)\}$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (B(X)|A(X))[p_1]$.

$$\Psi_{\mathcal{R}_7} \equiv_{\mathcal{S}} \forall X. \left[C(X)\Big(y_1 A(X)\left(x_1 B(X) \vee \overline{B(X)}\right) \vee \mathsf{s}(1)\overline{A(X)}\Big)\right]$$

$$\tau_{\mathcal{R}_7} = (y_1(x_1+1)+2)^m$$

Let $c \in \mathsf{Const}_\Sigma$.

$$\Psi_{\mathcal{R}_7} A(c) \equiv_{\mathcal{S}} \forall X_{\neq c}. \left[C(X)\Big(y_1 A(X)\left(x_1 B(X) \vee \overline{B(X)}\right) \vee \mathsf{s}(1)\overline{A(X)}\Big)\right]$$
$$y_1 A(c) C(c)\Big(x_1 B(c) \vee \overline{B(c)}\Big)$$
$$\Psi_{\mathcal{R}_7} A(c) B(c) \equiv_{\mathcal{S}} \forall X_{\neq c}. \left[C(X)\Big(y_1 A(X)\left(x_1 B(X) \vee \overline{B(X)}\right) \vee \mathsf{s}(1)\overline{A(X)}\Big)\right]$$
$$y_1 A(c) C(c)\Big(x_1 B(c)\Big)$$

$$\tau^{r_1}_{\mathcal{R}_7} = m\ (y_1(x_1+1)+2)^{m-1}\ (y_1 x_1 + z y_1(x_1+1))$$

▶ Knowledge Base $\mathcal{R}_8$

We consider $\mathsf{Pred}_\Sigma = \{A/1, B/1\}$, and $\mathcal{R}_8 = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \{\forall X.A(X)\}$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (B(X)|A(X))[p_1]$.

$$\Psi_{\mathcal{R}_8} \equiv_{\mathcal{S}} \forall X. \left[y_1 A(X)\left(x_1 B(X) \vee \overline{B(X)}\right)\right]$$

$$\tau_{\mathcal{R}_8} = y_1(x_1+1)^m$$

Let $c \in \mathsf{Const}_\Sigma$.

$$\begin{aligned}\Psi_{\mathcal{R}_8} A(c) &\equiv_{\mathcal{S}} \Psi_{\mathcal{R}_8}\\ \Psi_{\mathcal{R}_8} A(c)B(c) &\equiv_{\mathcal{S}} \forall X_{\neq c}.\left[y_1 A(X)\left(x_1 B(X) \vee \overline{B(X)}\right)\right]\\ &\quad y_1 A(c)\Big(x_1 B(c)\Big)\end{aligned}$$

$$\tau^{r_1}_{\mathcal{R}_8} = m\ y_1(x_1+1)^{m-1}\ (y_1 x_1 + z y_1(x_1+1))$$

▶ Knowledge Base $\mathcal{R}_9$

We consider $\mathsf{Pred}_\Sigma = \{A/1, B/1, C/1\}$, and $\mathcal{R}_9 = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \{\exists X.C(X)\}$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (B(X)|A(X))[p_1]$.

$$\Psi_{\mathcal{R}_9} \equiv_{\mathcal{S}} \Big[\exists X.C(X)\Big]\forall X.\left[y_1 A(X)\left(x_1 B(X) \vee \overline{B(X)}\right) \vee \overline{A(X)}\right]$$

Let S and Z be fresh predicates of arity 0.

$$\begin{aligned}\Psi^{\mathsf{skolem}}_{\mathcal{R}_9} &\equiv_{\mathcal{S}} Z\Big(S \vee (-1)\overline{S}\ \forall X.\overline{C(X)}\Big)\\ &\qquad \forall X.\left[y_1 A(X)\left(x_1 B(X) \vee \overline{B(X)}\right) \vee \overline{A(X)}\right]\\ &\equiv_{\mathcal{S}} Z\Big(\mathsf{s}(m)S \vee (-1)\overline{S}\ \forall X.\overline{C(X)}\Big)\\ &\qquad \forall X.\left[y_1 A(X)\left(x_1 B(X) \vee \overline{B(X)}\right) \vee \mathsf{s}(1)\overline{A(X)}\right]\end{aligned}$$

$$\tau_{\mathcal{R}_9} = (2^m - 1)\ (y_1(x_1+1)+2)^m$$

Let $c \in \mathsf{Const}_\Sigma$.

$$\begin{aligned}\Psi^{\mathsf{skolem}}_{\mathcal{R}_9} A(c) &\equiv_{\mathcal{S}} Z\Big(\mathsf{s}(m)S \vee (-1)\overline{S}\ \forall X.\overline{C(X)}\Big)\\ &\qquad \forall X_{\neq c}.\left[y_1 A(X)\left(x_1 B(X) \vee \overline{B(X)}\right) \vee \mathsf{s}(1)\overline{A(X)}\right]\\ &\qquad y_1 A(c)\Big(x_1 B(c) \vee \overline{B(c)}\Big)\\ \Psi^{\mathsf{skolem}}_{\mathcal{R}_9} A(c)B(c) &\equiv_{\mathcal{S}} Z\Big(\mathsf{s}(m)S \vee (-1)\overline{S}\ \forall X.\overline{C(X)}\Big)\\ &\qquad \forall X_{\neq c}.\left[y_1 A(X)\left(x_1 B(X) \vee \overline{B(X)}\right) \vee \mathsf{s}(1)\overline{A(X)}\right]\\ &\qquad y_1 A(c)\Big(x_1 B(c)\Big)\end{aligned}$$

$$\tau^{r_1}_{\mathcal{R}_9} = m\ (2^m - 1)\ (y_1(x_1+1)+2)^{m-1}\ (y_1x_1 + zy_1(x_1+1))$$

▶ Knowledge Base $\mathcal{R}_{10}$

We consider $\mathsf{Pred}_\Sigma = \{A/1, B/1\}$, and $\mathcal{R}_{10} = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \{\exists X.A(X)\}$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (B(X)|A(X))[p_1]$.

$$\Psi_{\mathcal{R}_{10}} \equiv_{\mathcal{S}} \Big[\exists X.A(X)\Big]\forall X.\ \Big[y_1A(X)\ \Big(x_1B(X) \lor \overline{B(X)}\Big) \lor \overline{A(X)}\Big]$$

Let S and Z be fresh predicates of arity 0.

$$\begin{aligned}
\Psi^{\mathsf{skolem}}_{\mathcal{R}_{10}} &\equiv_{\mathcal{S}} Z\Big(S \lor (-1)\overline{S}\ \forall X.\Big[\overline{A(X)}\Big]\Big)\forall X.\ \Big[y_1A(X)\ \Big(x_1B(X) \lor \overline{B(X)}\Big) \lor \overline{A(X)}\Big] \\
&\equiv_{\mathcal{S}} Z\Big(S\ \forall X.\ \Big[y_1A(X)\ \Big(x_1B(X) \lor \overline{B(X)}\Big) \lor \mathsf{s}(1)\overline{A(X)}\Big] \\
&\qquad \lor (-1)\mathsf{s}(m)\overline{S}\forall X.\Big[\overline{A(X)}\Big]\Big)
\end{aligned}$$

$$\tau_{\mathcal{R}_{10}} = (y_1(x_1+1)+2)^m - 2^m$$

Let $c \in \mathsf{Const}_\Sigma$.

$$\begin{aligned}
\Psi^{\mathsf{skolem}}_{\mathcal{R}_{10}}A(c) &\equiv_{\mathcal{S}} ZS\ \forall X_{\neq c}.\ \Big[y_1A(X)\ \Big(x_1B(X) \lor \overline{B(X)}\Big) \lor \mathsf{s}(1)\overline{A(X)}\Big] \\
&\qquad y_1A(c)\Big(x_1B(c) \lor \overline{B(c)}\Big) \\
\Psi^{\mathsf{skolem}}_{\mathcal{R}_{10}}A(c)B(c) &\equiv_{\mathcal{S}} ZS\ \forall X_{\neq c}.\ \Big[y_1A(X)\ \Big(x_1B(X) \lor \overline{B(X)}\Big) \lor \mathsf{s}(1)\overline{A(X)}\Big] \\
&\qquad y_1A(c)\Big(x_1B(c)\Big)
\end{aligned}$$

$$\tau^{r_1}_{\mathcal{R}_{10}} = m\ (y_1(x_1+1)+2)^{m-1}\ (y_1x_1 + zy_1(x_1+1))$$

▶ Knowledge Base $\mathcal{R}_{11}$

We consider $\mathsf{Pred}_\Sigma = \{A/1, R/2\}$, and $\mathcal{R}_{11} = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (R(X,Y)|A(X))[p_1]$.

$$\begin{aligned}\Psi_{\mathcal{R}_{11}} &\equiv_{\mathcal{S}} \forall X.\forall Y.\left[y_1 A(X)\left(x_1 R(X,Y) \vee \overline{R(X,Y)}\right) \vee \overline{A(X)}\right]\\ &\equiv_{\mathcal{S}} \forall X.\left[y_1^m A(X)\ \forall Y.\left[x_1 R(X,Y) \vee \overline{R(X,Y)}\right] \vee \mathsf{s}(m)\overline{A(X)}\right]\end{aligned}$$

$$\tau_{\mathcal{R}_{11}} = (y_1^m (x_1+1)^m + 2^m)^m$$

Let $c \in \mathsf{Const}_\Sigma$, and let $d \in \mathsf{Const}_\Sigma$ regardless of whether $c = d$ or not.

$$\begin{aligned}\Psi_{\mathcal{R}_{11}} A(c) &\equiv_{\mathcal{S}} \forall X_{\neq c}.\left[y_1^m A(X)\ \forall Y.\left[x_1 R(X,Y) \vee \overline{R(X,Y)}\right] \vee \mathsf{s}(m)\overline{A(X)}\right]\\ &\quad y_1^m A(c)\ \forall Y.\left[x_1 R(c,Y) \vee \overline{R(c,Y)}\right]\\ \Psi_{\mathcal{R}_{11}} A(c)R(c,d) &\equiv_{\mathcal{S}} \forall X_{\neq c}.\left[y_1^m A(X)\ \forall Y.\left[x_1 R(X,Y) \vee \overline{R(X,Y)}\right] \vee \mathsf{s}(m)\overline{A(X)}\right]\\ &\quad y_1^m A(c)\ \forall Y_{\neq d}.\left[x_1 R(c,Y) \vee \overline{R(c,Y)}\right] x_1 R(c,d)\end{aligned}$$

$$\tau^{r_1}_{\mathcal{R}_{11}} = m^2\ (y_1^m (x_1+1)^m + 2^m)^{m-1}\ y_1 (x_1+1)^{m-1}\ (y_1 x_1 + z y_1 (x_1+1))$$

▶ Knowledge Base $\mathcal{R}_{12}$

We consider $\mathsf{Pred}_\Sigma = \{R/2, S/2\}$, and $\mathcal{R}_{12} = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (S(X,Y)|R(X,Y))[p_1]$.

$$\Psi_{\mathcal{R}_{12}} \equiv_{\mathcal{S}} \forall X.\forall Y.\left[y_1 R(X,Y)\left(x_1 S(X,Y) \vee \overline{S(X,Y)}\right) \vee \mathsf{s}(1)\overline{R(X,Y)}\right]$$

$$\tau_{\mathcal{R}_{12}} = (y_1(x_1+1)+2)^{m^2}$$

Let $c \in \mathsf{Const}_\Sigma$, and let $d \in \mathsf{Const}_\Sigma$ regardless of whether $c = d$ or not.

$$\begin{aligned}
\Psi_{\mathcal{R}_{12}} R(c,d) \equiv_{\mathcal{S}} {} & \forall X_{\neq c}.\forall Y.\ \Big[y_1 R(X,Y)\ \Big(x_1 S(X,Y) \vee \overline{S(X,Y)} \Big) \vee \mathsf{s}(1)\overline{R(X,Y)} \Big] \\
& \forall Y_{\neq d}.\Big[y_1 R(c,Y)\Big(x_1 S(c,Y) \vee \overline{S(c,Y)} \Big) \vee \mathsf{s}(1)\overline{R(c,Y)} \Big] \\
& y_1 R(c,d)\Big(x_1 S(c,d) \vee \overline{S(c,d)} \Big)
\end{aligned}$$

$$\begin{aligned}
\Psi_{\mathcal{R}_{12}} R(c,d)S(c,d) \equiv_{\mathcal{S}} {} & \forall X_{\neq c}.\forall Y.\ \Big[y_1 R(X,Y)\ \Big(x_1 S(X,Y) \vee \overline{S(X,Y)} \Big) \vee \mathsf{s}(1)\overline{R(X,Y)} \Big] \\
& \forall Y_{\neq d}.\Big[y_1 R(c,Y)\Big(x_1 S(c,Y) \vee \overline{S(c,Y)} \Big) \vee \mathsf{s}(1)\overline{R(c,Y)} \Big] \\
& y_1 R(c,d)\Big(x_1 S(c,d) \Big)
\end{aligned}$$

$$\boxed{\tau^{r_1}_{\mathcal{R}_{12}} = m^2\ \left(y_1(x_1+1)+2\right)^{m^2-1}\ \left(y_1 x_1 + z y_1(x_1+1)\right)}$$

▶ Knowledge Base $\mathcal{R}_{13}$

We consider $\mathsf{Pred}_\Sigma = \{R/2, S/2\}$, and $\mathcal{R}_{13} = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (S(Y,X)|R(X,Y))[p_1]$.

$$\Psi_{\mathcal{R}_{13}} \equiv_{\mathcal{S}} \forall X.\forall Y.\ \Big[y_1 R(X,Y)\ \Big(x_1 S(Y,X) \vee \overline{S(Y,X)} \Big) \vee \mathsf{s}(1)\overline{R(X,Y)} \Big]$$

$$\boxed{\tau_{\mathcal{R}_{13}} = \left(y_1(x_1+1)+2\right)^{m^2}}$$

Let $c \in \mathsf{Const}_\Sigma$, and let $d \in \mathsf{Const}_\Sigma$ regardless of whether $c = d$ or not.

$$\begin{aligned}
\Psi_{\mathcal{R}_{13}} R(c,d) \equiv_{\mathcal{S}} {} & \forall X_{\neq c}.\forall Y.\ \Big[y_1 R(X,Y)\ \Big(x_1 S(Y,X) \vee \overline{S(Y,X)} \Big) \vee \mathsf{s}(1)\overline{R(X,Y)} \Big] \\
& \forall Y_{\neq d}.\Big[y_1 R(c,Y)\Big(x_1 S(Y,c) \vee \overline{S(Y,c)} \Big) \vee \mathsf{s}(1)\overline{R(c,Y)} \Big] \\
& y_1 R(c,d)\Big(x_1 S(d,c) \vee \overline{S(d,c)} \Big)
\end{aligned}$$

$$\begin{aligned}
\Psi_{\mathcal{R}_{12}} R(c,d)S(d,c) \equiv_{\mathcal{S}} {} & \forall X_{\neq c}.\forall Y.\ \Big[y_1 R(X,Y)\ \Big(x_1 S(Y,X) \vee \overline{S(Y,X)} \Big) \vee \mathsf{s}(1)\overline{R(X,Y)} \Big] \\
& \forall Y_{\neq d}.\Big[y_1 R(c,Y)\Big(x_1 S(Y,c) \vee \overline{S(Y,c)} \Big) \vee \mathsf{s}(1)\overline{R(c,Y)} \Big] \\
& y_1 R(c,d)\Big(x_1 S(d,c) \Big)
\end{aligned}$$

$$\tau^{r_1}_{\mathcal{R}_{12}} = m^2\ (y_1(x_1+1)+2)^{m^2-1}\ (y_1x_1 + zy_1(x_1+1))$$

▶ Knowledge Base $\mathcal{R}_{14}$

We consider $\mathsf{Pred}_\Sigma = \{A/1, B/1\}$, and $\mathcal{R}_{14} = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (B(X)|A(X)\overline{A(Y)})[p_1]$.

$m = 1$

The knowledge base $\mathcal{R}_{14}$ is inconsistent.

$m > 1$

We make use of set disjunction.

$$\begin{aligned}\Psi_{\mathcal{R}_{14}} &\equiv_{\mathcal{S}} \forall X.\forall Y. \left[y_1 A(X)\overline{A(Y)}\left(x_1B(X)\vee\overline{B(X)}\right)\vee\overline{A(X)}\vee A(Y)\right]\\ &\equiv_{\mathcal{S}} \bigvee_{\mathcal{A}\subseteq\mathsf{Const}_\Sigma}\Bigg(\mathsf{s}(m-|\mathcal{A}|)\ \forall X_{\in\mathcal{A}}.\left[A(X)\right]\forall X_{\in\mathsf{Const}_\Sigma\setminus\mathcal{A}}.\left[\overline{A(X)}\right]\\ &\qquad \forall X_{\in\mathcal{A}}.\left[y_1^{m-|\mathcal{A}|}\left(x_1^{m-|\mathcal{A}|}B(X)\vee\overline{B(X)}\right)\right]\Bigg)\end{aligned}$$

$$\tau_{\mathcal{R}_{14}} = \sum_{k=0}^{m}\binom{m}{k}\ 2^{m-k}\ (y_1^{m-k}(x_1^{m-k}+1))^k$$

Let $c \in \mathsf{Const}_\Sigma$.

$$\Psi_{\mathcal{R}_{14}}A(c)\overline{A(c)} \equiv_{\mathcal{S}} \Psi_{\mathcal{R}_{14}}A(c)\overline{A(c)}B(c) \equiv_{\mathcal{S}} \bot$$

Let $d \in \mathsf{Const}_\Sigma$ with $c \neq d$.

$$\begin{aligned}\Psi_{\mathcal{R}_{14}}A(c)\overline{A(d)} &\equiv_{\mathcal{S}} \mathsf{s}(1)y_1A(c)\overline{A(d)} \bigvee_{\mathcal{A}\subseteq\mathsf{Const}_\Sigma\setminus\{c,d\}}\Bigg(\left(x_1^{|\mathcal{A}^C|+1}B(c)\vee\overline{B(c)}\right)\\ &\qquad \mathsf{s}(|\mathcal{A}^C|)y_1^{|\mathcal{A}^C|(|\mathcal{A}|+1)+1}\forall X_{\in\mathcal{A}}.\left[A(X)\right]\forall X_{\in\mathcal{A}^C}.\left[\overline{A(X)}\right]\\ &\qquad \forall X_{\in\mathcal{A}}.\left[x_1^{|\mathcal{A}^C|+1}B(X)\vee\overline{B(X)}\right]\Bigg)\\ \Psi_{\mathcal{R}_{14}}A(c)\overline{A(d)}B(c) &\equiv_{\mathcal{S}} \mathsf{s}(1)y_1A(c)\overline{A(d)} \bigvee_{\mathcal{A}\subseteq\mathsf{Const}_\Sigma\setminus\{c,d\}}\Bigg(\left(x_1^{|\mathcal{A}^C|+1}B(c)\right)\\ &\qquad \mathsf{s}(|\mathcal{A}^C|)y_1^{|\mathcal{A}^C|(|\mathcal{A}|+1)+1}\forall X_{\in\mathcal{A}}.\left[A(X)\right]\forall X_{\in\mathcal{A}^C}.\left[\overline{A(X)}\right]\end{aligned}$$

$$\forall X_{\in\mathcal{A}}.\Big[x_1^{|\mathcal{A}^C|+1}B(X)\vee\overline{B(X)}\Big]\Big)$$

where $\mathcal{A}^C=(\mathsf{Const}_\Sigma\setminus\{c,d\})\setminus\mathcal{A}$.

$$\tau^{r_1}_{\mathcal{R}_{14}}=m\;(m-1)\sum_{k=0}^{m-2}\binom{m-2}{k}((1+z)x_1^{m-k-1}+1)\;2^{m-k-1}\\ y_1^{m(k+1)+k(k-3)}\;(x_1^{m-k-1}+1)^k$$

▶ Knowledge Base $\mathcal{R}_{15}$

We consider $\mathsf{Pred}_\Sigma=\{A/1,R/2\}$, and $\mathcal{R}_{15}=(\mathcal{F},\mathcal{B})$ with

- $\mathcal{F}=\emptyset$,
- $\mathcal{B}=\{r_1\}$ with $r_1=(A(Y)|A(X)\wedge R(X,Y))[p_1]$.

$m=1$

Let $c\in\mathsf{Const}_\Sigma$.

$$\Psi_{\mathcal{R}_{15}}\equiv_{\mathcal{S}}A(c)\Big(y_1x_1R(c,c)\vee\overline{R(c,c)}\Big)\vee\mathsf{s}(1)\overline{A(c)}$$

$$\tau_{\mathcal{R}_{15}}=y_1x_1+3$$

$$\Psi_{\mathcal{R}_{15}}A(c)R(c,c)\equiv_{\mathcal{S}}\Psi_{\mathcal{R}_{15}}A(c)R(c,c)A(c)\equiv_{\mathcal{S}}y_1x_1$$

$$\tau^{r_1}_{\mathcal{R}_{15}}=y_1x_1(1+z)$$

$m>1$

We make use of set disjunction.

$$\begin{aligned}\Psi_{\mathcal{R}_{15}}&\equiv_{\mathcal{S}}\forall X.\forall Y.\Big[y_1A(X)R(X,Y)\,\Big(x_1A(Y)\vee\overline{A(Y)}\Big)\vee\overline{A(X)}\vee\overline{R(X,Y)}\Big]\\&\equiv_{\mathcal{S}}\bigvee_{\mathcal{A}\subseteq\mathsf{Const}_\Sigma}\Big(\mathsf{s}((m-|\mathcal{A}|)\;m)\;\forall X_{\in\mathcal{A}}.\Big[A(X)\Big]\forall X_{\in\mathsf{Const}_\Sigma\setminus\mathcal{A}}.\Big[\overline{A(X)}\Big]\\&\qquad y_1^{m|\mathcal{A}|}x_1^{|\mathcal{A}|^2}\forall X_{\in\mathcal{A}}.\forall Y.\Big[R(X,Y)\Big]\Big)\end{aligned}$$

$$\tau_{\mathcal{R}_{15}} = \sum_{k=0}^{m} \binom{m}{k} \, 2^{m(m-k)} \; y_1^{mk} \; x_1^{k^2}$$

Let $c \in \mathsf{Const}_\Sigma$.

$$\begin{aligned}
\Psi_{\mathcal{R}_{15}} A(c) R(c,c) &\equiv_{\mathcal{S}} \Psi_{\mathcal{R}_{15}} A(c) R(c,c) A(c) \\
&\equiv_{\mathcal{S}} y_1 x_1 A(c) R(c,c) \bigvee_{\mathcal{A} \subseteq \mathsf{Const}_\Sigma \setminus \{c\}} \Big(\forall X_{\in \mathcal{A}}. \Big[A(X) \Big] \forall_{X \in \mathcal{A}^C}. \Big[\overline{A(C)} \Big] \\
&\qquad \mathsf{s}(|\mathcal{A}^C| \; m) \; \forall X_{\in \mathcal{A}}. \Big[\forall Y_{\in \mathcal{A}}. \Big[y_1 x_1 R(X,Y) \vee \overline{R(X,Y)} \Big] \\
&\qquad \forall Y_{\in \mathcal{A}^C}. \Big[y_1 R(X,Y) \vee \overline{R(X,Y)} \Big] \Big(y_1 x_1 R(X,c) \vee \overline{R(X,c)} \Big) \Big] \\
&\qquad \forall Y_{\in \mathcal{A}}. \Big[y_1 x_1 R(c,Y) \vee \overline{R(c,Y)} \Big] \forall Y_{\in \mathcal{A}^C}. \Big[y_1 R(c,Y) \vee \overline{R(c,Y)} \Big] \Big)
\end{aligned}$$

where $\mathcal{A}^C = (\mathsf{Const}_\Sigma \setminus \{c\}) \setminus \mathcal{A}$. Let $d \in \mathsf{Const}_\Sigma$ with $d \neq c$.

$$\begin{aligned}
\Psi_{\mathcal{R}_{15}} A(c) R(c,d) &\equiv_{\mathcal{S}} A(c) R(c,d) \bigvee_{\mathcal{A} \subseteq \mathsf{Const}_\Sigma \setminus \{c,d\}} \Big(\forall X_{\in \mathcal{A}}. \Big[A(X) \Big] \forall X_{\in \mathcal{A}^C}. \Big[\overline{A(X)} \Big] \\
&\qquad y_1 \; \forall X_{\in \mathcal{A}}. \Big[\forall Y_{\in \mathcal{A} \cup \{c\}}. \Big[y_1 x_1 R(X,Y) \vee \overline{R(X,Y)} \Big] \\
&\qquad\qquad \forall Y_{\in \mathcal{A}^C}. \Big[y_1 R(X,Y) \vee \overline{R(X,Y)} \Big] \Big] \\
&\qquad \forall Y_{\in \mathcal{A}}. \; \Big[y_1 x_1 R(c,Y) \vee \overline{R(c,Y)} \Big] \; \forall Y_{\in \mathcal{A}^C}. \; \Big[y_1 R(c,Y) \vee \overline{R(c,Y)} \Big] \\
&\qquad \Big(\overline{A(d)} \; \mathsf{s}(m) \; \Big(y_1 R(c,c) \vee \overline{R(c,c)} \Big) \forall X_{\in \mathcal{A}}. \; \Big[y_1 R(X,d) \vee \overline{R(X,d)} \Big] \\
&\qquad \vee A(d) \Big(y_1 x_1 R(c,c) \vee \overline{R(c,c)} \Big) \forall X_{\in \mathcal{A}}. \; \Big[y_1 x_1 R(X,d) \vee \overline{R(X,d)} \Big] \\
&\qquad \forall Y_{\in \mathcal{A}}. \; \Big[y_1 x_1 R(d,Y) \vee \overline{R(d,Y)} \Big] \; \forall Y_{\in \mathcal{A}^C}. \; \Big[y_1 R(d,Y) \vee \overline{R(d,Y)} \Big] \\
&\qquad x_1 \Big(y_1 R(d,c) \vee \overline{R(d,c)} \Big) \Big(y_1 x_1 R(d,d) \vee \overline{R(d,d)} \Big) \Big) \Big)
\end{aligned}$$

$$\begin{aligned}
\Psi_{\mathcal{R}_{15}} A(c) R(c,d) A(d) &\equiv_{\mathcal{S}} A(c) R(c,d) \bigvee_{\mathcal{A} \subseteq \mathsf{Const}_\Sigma \setminus \{c,d\}} \Big(\forall X_{\in \mathcal{A}}. \Big[A(X) \Big] \forall X_{\in \mathcal{A}^C}. \Big[\overline{A(X)} \Big] \\
&\qquad y_1 \; \forall X_{\in \mathcal{A}}. \Big[\forall Y_{\in \mathcal{A} \cup \{c\}}. \Big[y_1 x_1 R(X,Y) \vee \overline{R(X,Y)} \Big] \\
&\qquad\qquad \forall Y_{\in \mathcal{A}^C}. \Big[y_1 R(X,Y) \vee \overline{R(X,Y)} \Big] \Big] \\
&\qquad \forall Y_{\in \mathcal{A}}. \; \Big[y_1 x_1 R(c,Y) \vee \overline{R(c,Y)} \Big] \; \forall Y_{\in \mathcal{A}^C}. \; \Big[y_1 R(c,Y) \vee \overline{R(c,Y)} \Big] \\
&\qquad \Big(y_1 x_1 R(c,c) \vee \overline{R(c,c)} \Big) \forall X_{\in \mathcal{A}}. \; \Big[y_1 x_1 R(X,d) \vee \overline{R(X,d)} \Big] \\
&\qquad \forall Y_{\in \mathcal{A}}. \; \Big[y_1 x_1 R(d,Y) \vee \overline{R(d,Y)} \Big] \; \forall Y_{\in \mathcal{A}^C}. \; \Big[y_1 R(d,Y) \vee \overline{R(d,Y)} \Big] \\
&\qquad x_1 \Big(y_1 R(d,c) \vee \overline{R(d,c)} \Big) \Big(y_1 x_1 R(d,d) \vee \overline{R(d,d)} \Big) \Big)
\end{aligned}$$

$$\begin{aligned}
\tau^{r_1}_{\mathcal{R}_{15}} &= (1+z)\ m y_1 x_1 \sum_{k=0}^{m-1} \binom{m-1}{k} 2^{(m-1-k)m} (y_1x_1+1)^{k(k+2)} (y_1+1)^{(m-1-k)(k+1)} \\
&\quad + m(m-1)\ \sum_{k=0}^{m-2} \binom{m-2}{k} y_1x_1(y_1x_1+1)^{k(k+4)+2}(y_1+1)^{(m-k-2)(k+2)+1} \\
&\quad + z\ m(m-1)\ \sum_{k=0}^{m-2} \binom{m-2}{k} y_1(y_1x_1+1)^{k(k+2)}(y_1+1)^{(m-k-1)(k+1)} \\
&\quad (2^m + x_1(y_1x_1+1)^{2k+2}(y_1+1)^{m-2k-2}) \\
&= (1+z)my_1x_1(y_1x_1+1)^{(m-1)(m+1)} + \\
&\quad my_1 \sum_{k=0}^{m-2}(y_1x_1+1)^{k(k+2)}(y_1+1)^{(m-1-k)(k+1)} \Big((1+z)x_1\binom{m-1}{k}2^{(m-1-k)m} \\
&\quad + (m-1)x_1\binom{m-2}{k}(y_1x_1+1)^{2k+2}(y_1+1)^{m-2k} + z(m-1)\binom{m-2}{k} \\
&\quad (2^m + x_1(y_1x_1+1)^{2k+2}(y_1+1)^{m-2k-2})\Big)
\end{aligned}$$

▶ Knowledge Base $\mathcal{R}_{16}$

We consider $\mathsf{Pred}_\Sigma = \{R/2\}$, and $\mathcal{R}_{16} = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (R(Y,X)|R(X,Y))[p_1]$.

$m = 1$

Let $c \in \mathsf{Const}_\Sigma$.

$$\begin{aligned}
\Psi_{\mathcal{R}_{16}} &\equiv_{\mathcal{S}} y_1 R(c,c)\Big(x_1 R(c,c) \vee \overline{R(c,c)}\Big) \vee \overline{R(c,c)} \\
&\equiv_{\mathcal{S}} y_1 x_1 R(c,c) \vee \overline{R(c,c)}
\end{aligned}$$

$$\tau_{\mathcal{R}_{16}} = y_1x_1 + 1$$

$$\Psi_{\mathcal{R}_{16}} R(c,c) \equiv_{\mathcal{S}} y_1 x_1 R(c,c)$$

$$\tau^{r_1}_{\mathcal{R}_{16}} = y_1x_1(1+z)$$

$m = 2$

Let $c \in \mathsf{Const}_\Sigma$, and let $d \in \mathsf{Const}_\Sigma$ with $d \neq c$.

$$\begin{aligned}\Psi_{\mathcal{R}_{16}} \equiv_{\mathcal{S}} {} & \forall X_{\in\{c,d\}}.\Big[y_1x_1R(X,X) \vee \overline{R(X,X)}\Big] \\ & \Big(y_1R(c,d)\Big(y_1x_1^2R(d,c) \vee \overline{R(d,c)}\Big) \vee \overline{R(c,d)}\Big(y_1R(d,c) \vee \overline{R(d,c)}\Big)\Big)\end{aligned}$$

$$\tau_{\mathcal{R}_{16}} = (y_1x_1+1)^2(y_1(y_1x_1^2+1)+y_1+1)$$

$$\begin{aligned}\Psi_{\mathcal{R}_{16}}R(c,c) \equiv_{\mathcal{S}} {} & \Big(y_1x_1R(d,d) \vee \overline{R(d,d)}\Big)y_1x_1R(c,c) \\ & \Big(y_1R(c,d)\Big(y_1x_1^2R(d,c) \vee \overline{R(d,c)}\Big) \vee \overline{R(c,d)}\Big(y_1R(d,c) \vee \overline{R(d,c)}\Big)\Big) \\ \Psi_{\mathcal{R}_{16}}R(c,d) \equiv_{\mathcal{S}} {} & \forall X_{\in\{c,d\}}.\Big[y_1x_1R(X,X) \vee \overline{R(X,X)}\Big] \\ & y_1R(c,d)\Big(y_1x_1^2R(d,c) \vee \overline{R(d,c)}\Big) \\ \Psi_{\mathcal{R}_{16}}R(c,d)R(d,c) \equiv_{\mathcal{S}} {} & \forall X_{\in\{c,d\}}.\Big[y_1x_1R(X,X) \vee \overline{R(X,X)}\Big] \\ & y_1R(c,d)\Big(y_1x_1^2R(d,c)\Big)\end{aligned}$$

$$\begin{aligned}\tau^{r_1}_{\mathcal{R}_{16}} &= 2(y_1x_1+1)y_1x_1(y_1(y_1x_1^2+1)+y_1+1)(1+z) \\ &\quad + 2(y_1x_1+1)^2y_1((y_1x_1^2)(z+1)+1) \\ &= 2(y_1x_1+1)y_1(x_1(y_1(y_1x_1^2+1)+y_1+1)(1+z) \\ &\quad + (y_1x_1+1)((y_1x_1^2)(z+1)+1)\end{aligned}$$

$m > 2$

$$\begin{aligned}\Psi_{\mathcal{R}_{16}} &\equiv_{\mathcal{S}} \forall X.\forall Y.\Big[y_1R(X,Y)\Big(x_1R(Y,X) \vee \overline{R(Y,X)}\Big) \vee \overline{R(X,Y)}\Big] \\ &\equiv_{\mathcal{S}} \forall X.\Big[y_1x_1R(X,X) \vee \overline{R(X,X)}\Big] \\ &\qquad \forall X_{X<Y}.\forall Y.\Big[y_1R(X,Y)\Big(y_1x_1^2R(Y,X) \vee \overline{R(Y,X)}\Big) \\ &\qquad \vee \overline{R(X,Y)}\Big(y_1R(Y,X) \vee \overline{R(Y,X)}\Big)\Big]\end{aligned}$$

$$\tau_{\mathcal{R}_{16}} = (y_1x_1+1)^m \; \left(y_1(y_1x_1^2+1)+(y_1+1)\right)^{\frac{m(m-1)}{2}}$$

Let $c \in \mathsf{Const}_\Sigma$.

$$\begin{aligned}
\Psi_{\mathcal{R}_{16}} R(c,c) \equiv_{\mathcal{S}} \; & \forall X_{\neq c}.\Big[y_1x_1R(X,X) \vee \overline{R(X,X)}\Big] y_1x_1R(c,c)\\
& \forall X_{X<Y}.\forall Y.\Big[y_1R(X,Y)\Big(y_1x_1^2R(Y,X) \vee \overline{R(Y,X)}\Big)\\
& \vee \overline{R(X,Y)}\Big(y_1R(Y,X) \vee \overline{R(Y,X)}\Big)\Big]
\end{aligned}$$

Let $d \in \mathsf{Const}_\Sigma$ with $d \neq c$.

$$\begin{aligned}
\Psi_{\mathcal{R}_{16}} R(c,d) \equiv_{\mathcal{S}} \; & \forall X.\Big[y_1x_1R(X,X) \vee \overline{R(X,X)}\Big]\\
& \forall X_{\neq c,d;X<Y}.\forall Y_{\neq c,d}.\Big[y_1R(X,Y)\Big(y_1x_1^2R(Y,X) \vee \overline{R(Y,X)}\Big)\\
& \vee \overline{R(X,Y)}\Big(y_1R(Y,X) \vee \overline{R(Y,X)}\Big)\Big]\\
& \forall X_{\neq c,d}.\forall Y_{\in\{c,d\}}.\Big[y_1R(X,Y)\Big(y_1x_1^2R(Y,X) \vee \overline{R(Y,X)}\Big)\\
& \vee \overline{R(X,Y)}\Big(y_1R(Y,X) \vee \overline{R(Y,X)}\Big)\Big]\\
& y_1R(c,d)\Big(y_1x_1^2R(d,c) \vee \overline{R(d,c)}\Big)\\
\Psi_{\mathcal{R}_{16}} R(c,d)R(d,c) \equiv_{\mathcal{S}} \; & \forall X.\Big[y_1x_1R(X,X) \vee \overline{R(X,X)}\Big]\\
& \forall X_{\neq c,d;X<Y}.\forall Y_{\neq c,d}.\Big[y_1R(X,Y)\Big(y_1x_1^2R(Y,X) \vee \overline{R(Y,X)}\Big)\\
& \vee \overline{R(X,Y)}\Big(y_1R(Y,X) \vee \overline{R(Y,X)}\Big)\Big]\\
& \forall X_{\neq c,d}.\forall Y_{\in\{c,d\}}.\Big[y_1R(X,Y)\Big(y_1x_1^2R(Y,X) \vee \overline{R(Y,X)}\Big)\\
& \vee \overline{R(X,Y)}\Big(y_1R(Y,X) \vee \overline{R(Y,X)}\Big)\Big]\\
& y_1R(c,d)\Big(y_1x_1^2R(d,c)\Big)
\end{aligned}$$

$$\begin{aligned}
\tau^{r_1}_{\mathcal{R}_{16}} &= m(y_1x_1+1)^{m-1}(y_1(y_1x_1^2+1)+(y_1+1))^{\frac{m(m-1)}{2}}(1+z)\\
&\quad + m(m-1)(y_1x_1+1)^m \; (y_1(y_1x_1^2+1)+(y_1+1))^{\frac{(m-2)(m-3)}{2}}\\
&\quad (y_1(y_1x_1^2+1)+(y_1+1))^{2(m-2)}y_1(y_1x_1^2+1+zy_1x_1^2)\\
&= m(y_1x_1+1)^{m-1}(y_1(y_1x_1^2+1)+(y_1+1))^{\frac{(m-2)(m+1)}{2}}\\
&\quad ((y_1(y_1x_1^2+1)+(y_1+1))(1+z)+(m-1)(y_1x_1+1)(y_1x_1^2(1+z)+1))
\end{aligned}$$

▶ Knowledge Base $\mathcal{R}_{17}$

We consider $\mathsf{Pred}_\Sigma = \{R/2\}$, and $\mathcal{R}_{17} = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (R(X,Z) | R(X,Y) \wedge R(Y,Z))[p_1]$.

Determining $\Psi_{\mathcal{R}_{17}}$ is hard since counting transitive relations is a difficult problem (cf. [Mala, 2022; Pfeiffer, 2004]). The method of choice so far would be completely grounding the structured sentence.

▶ Knowledge Base $\mathcal{R}_{18}$

We consider $\mathsf{Pred}_\Sigma = \{A/1, R/2\}$, and $\mathcal{R}_{18} = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (\forall Y. R(X,Y) \Rightarrow A(Y) | A(X))[p_1]$.

$$\Psi_{\mathcal{R}_{18}} \equiv_{\mathcal{S}} \forall X. \Big[y_1 A(X) \Big(x_1 \forall Y. \Big[\overline{R(X,Y)} \vee A(Y) \Big] \vee \exists Y. \Big[R(X,Y) \overline{A(Y)} \Big] \Big) \vee \overline{A(X)} \Big]$$

Let S and Z be fresh predicates of arity 1.

$$\begin{aligned}
\Psi^{\mathsf{skolem}}_{\mathcal{R}_{18}} \equiv_{\mathcal{S}}\ & \forall X. \Big[y_1 A(X) \Big(x_1 \overline{Z(X)} \vee Z(X) \Big) \vee \overline{A(X)} \Big] \\
& \forall X. \Big[Z(X) \Big(S(X) \vee (-1) \overline{S(X)} \forall Y. \Big[\overline{R(X,Y)} \vee A(Y) \Big] \Big) \\
& \vee \overline{Z(X)} S(X) \forall Y. \Big[\overline{R(X,Y)} \vee A(Y) \Big] \Big] \\
\equiv_{\mathcal{S}}\ & \bigvee_{\mathcal{A} \subseteq \mathsf{Const}_\Sigma} \Big(\forall X_{\in \mathcal{A}}. \Big[A(X) \Big] \forall X \in \mathcal{A}^c. \Big[\overline{A(X)} \Big] \mathsf{s}(|\mathcal{A}| \cdot m)\ y_1^{|\mathcal{A}|} \\
& \forall X_{\in \mathcal{A}}. \Big[Z(X) \Big(\mathsf{s}(|\mathcal{A}^c|) S(X) \vee (-1) \overline{S(X)} \forall Y_{\in \mathcal{A}^c}. \Big[\overline{R(X,Y)} \Big] \Big) \\
& \vee x_1 \overline{Z(X)} S(X) \forall Y_{\in \mathcal{A}^c}. \Big[\overline{R(X,Y)} \Big] \Big] \\
& \forall X_{\in \mathcal{A}^c}. \Big[Z(X) \Big(\mathsf{s}(|\mathcal{A}^c|) S(X) \vee (-1) \overline{S(X)} \forall Y_{\in \mathcal{A}^c}. \Big[\overline{R(X,Y)} \Big] \Big) \\
& \vee \overline{Z(X)} S(X) \forall Y_{\in \mathcal{A}^c}. \Big[\overline{R(X,Y)} \Big] \Big] \Big)
\end{aligned}$$

$$\tau_{\mathcal{R}_{18}} = \sum_{k=0}^{m} \binom{m}{k} 2^{k \cdot m}\ y_1^k\ (2^{m-k} - 1 + x_1)^k\ 2^{(m-k)^2}$$

Let $c \in \mathsf{Const}_\Sigma$.

$$\begin{aligned}\Psi^{\mathsf{skolem}}_{\mathcal{R}_{18}} A(c) \equiv_{\mathcal{S}} \bigvee_{\mathcal{A}\subseteq\mathsf{Const}_\Sigma\setminus\{c\}} & \Big(A(c)\ \forall X_{\in\mathcal{A}}.\Big[A(X)\Big]\forall X\in \mathcal{A}^c.\Big[\overline{A(X)}\Big]\mathsf{s}((|\mathcal{A}|+1)\cdot m)\ y_1^{|\mathcal{A}|+1}\\ & \forall X_{\in\mathcal{A}\cup\{c\}}.\Big[Z(X)\Big(\mathsf{s}(|\mathcal{A}^c|)S(X)\vee(-1)\overline{S(X)}\forall Y_{\in\mathcal{A}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\\ & \vee x_1\overline{Z(X)}S(X)\forall Y_{\in\mathcal{A}^c}.\Big[\overline{R(X,Y)}\Big]\Big]\\ & \forall X_{\in\mathcal{A}^c}.\Big[Z(X)\Big(\mathsf{s}(|\mathcal{A}^c|)S(X)\vee(-1)\overline{S(X)}\forall Y_{\in\mathcal{A}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\\ & \vee\overline{Z(X)}S(X)\forall Y_{\in\mathcal{A}^c}.\Big[\overline{R(X,Y)}\Big]\Big]\Big)\end{aligned}$$

$$\Psi^{\mathsf{skolem}}_{\mathcal{R}_{18}} A(c)\forall Y.\Big[\overline{R(c,Y)}\vee A(Y)\Big]\equiv_{\mathcal{S}}$$

$$\begin{aligned}\bigvee_{\mathcal{A}\subseteq\mathsf{Const}_\Sigma\setminus\{c\}} & \Big(A(c)\ \forall X_{\in\mathcal{A}}.\Big[A(X)\Big]\forall X\in \mathcal{A}^c.\Big[\overline{A(X)}\Big]\mathsf{s}((|\mathcal{A}|+1)\cdot m)\ y_1^{|\mathcal{A}|+1}\\ & \forall X_{\in\mathcal{A}}.\Big[Z(X)\Big(\mathsf{s}(|\mathcal{A}^c|)S(X)\vee(-1)\overline{S(X)}\forall Y_{\in\mathcal{A}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\\ & \vee x_1\overline{Z(X)}S(X)\forall Y_{\in\mathcal{A}^c}.\Big[\overline{R(X,Y)}\Big]\Big]\\ & \forall X_{\in\mathcal{A}^c}.\Big[Z(X)\Big(\mathsf{s}(|\mathcal{A}^c|)S(X)\vee(-1)\overline{S(X)}\forall Y_{\in\mathcal{A}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\\ & \vee\overline{Z(X)}S(X)\forall Y_{\in\mathcal{A}^c}.\Big[\overline{R(X,Y)}\Big]\Big]\\ & \Big[Z(c)\Big(S(c)\vee(-1)\overline{S(c)}\Big)\vee x_1\overline{Z(c)}S(c)\Big]\forall Y_{\in\mathcal{A}^c}.\Big[\overline{R(c,Y)}\Big]\Big)\end{aligned}$$

$$\begin{aligned}\tau^{r_1}_{\mathcal{R}_{18}} = m\sum_{k=0}^{m-1}\binom{m-1}{k}2^{(k+1)m}\ y_1^{k+1}\\ (2^{m-1-k}-1+x_1)^k\ 2^{(m-1-k)^2}\ (x_1+z(2^{m-1-k}-1+x_1))\end{aligned}$$

▶ Knowledge Base $\mathcal{R}_{19}$

We consider $\mathsf{Pred}_\Sigma = \{A/1, B/1, R/2\}$, and $\mathcal{R}_{21} = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (\forall Y.R(X,Y) \Rightarrow B(Y)|A(X))[p_1]$.

$$\Psi_{\mathcal{R}_{19}} \equiv_{\mathcal{S}} \forall X.\Big[y_1 A(X)\Big(x_1\forall Y.\Big[\overline{R(X,Y)}\vee B(Y)\Big]\vee\exists Y.\Big[R(X,Y)\overline{B(Y)}\Big]\Big)\vee\overline{A(X)}\Big]$$

Let S and Z be fresh predicates of arity 1.

$$\begin{aligned}\Psi^{\mathsf{skolem}}_{\mathcal{R}_{19}} \equiv_{\mathcal{S}} & \forall X.\Big[y_1 A(X)\Big(x_1\overline{Z(X)}\vee Z(X)\Big)\vee\overline{A(X)}\Big]\\ & \forall X.\Big[Z(X)\Big(S(X)\vee(-1)\overline{S(X)}\forall Y.\Big[\overline{R(X,Y)}\vee B(Y)\Big]\Big)\end{aligned}$$

$$
\begin{aligned}
&\qquad \vee\ \overline{Z(X)}S(X)\forall Y.\Big[\overline{R(X,Y)} \vee B(Y)\Big]\Big]\\
&\equiv_{\mathcal{S}} \bigvee_{\mathcal{B}\subseteq \mathsf{Const}_\Sigma} \Big(\forall X_{\in\mathcal{B}}.\Big[B(X)\Big]\forall X_{\in\mathcal{B}^c}.\Big[\overline{B(X)}\Big]\mathsf{s}(|\mathcal{B}|\cdot m)\\
&\qquad \forall X.\Big[y_1A(X)\Big(Z(X)\Big(\mathsf{s}(|\mathcal{B}^c|)S(X) \vee (-1)\overline{S(X)}\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\\
&\qquad \vee\ x_1\overline{Z(X)}S(X)\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\\
&\qquad \vee\ \overline{A(X)}\Big(Z(X)\Big(\mathsf{s}(|\mathcal{B}^c|)S(X) \vee (-1)\overline{S(X)}\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\\
&\qquad \vee\ \overline{Z(X)}S(X)\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\Big]\Big)
\end{aligned}
$$

$$
\tau_{\mathcal{R}_{18}} = \sum_{k=0}^{m}\binom{m}{k}2^{k\cdot m}\ (y_1(2^{m-k}-1+x_1)+2^{m-k})^m
$$

Let $c \in \mathsf{Const}_\Sigma$.

$$
\begin{aligned}
\Psi^{\mathsf{skolem}}_{\mathcal{R}_{19}}A(c) \equiv_{\mathcal{S}} &\bigvee_{\mathcal{B}\subseteq \mathsf{Const}_\Sigma} \Big(\forall X_{\in\mathcal{B}}.\Big[B(X)\Big]\forall X_{\in\mathcal{B}^c}.\Big[\overline{B(X)}\Big]\mathsf{s}(|\mathcal{B}|\cdot m)\\
&\forall X_{\neq c}.\Big[y_1A(X)\Big(Z(X)\Big(\mathsf{s}(|\mathcal{B}^c|)S(X) \vee (-1)\overline{S(X)}\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\\
&\vee\ x_1\overline{Z(X)}S(X)\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\\
&\vee\ \overline{A(X)}\Big(Z(X)\Big(\mathsf{s}(|\mathcal{B}^c|)S(X) \vee (-1)\overline{S(X)}\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\\
&\vee\ \overline{Z(X)}S(X)\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\Big]\\
&y_1A(c)\Big(Z(c)\Big(\mathsf{s}(|\mathcal{B}^c|)S(c) \vee (-1)\overline{S(c)}\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(c,Y)}\Big]\Big)\\
&\vee\ x_1\overline{Z(c)}S(c)\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(c,Y)}\Big]\Big)\Big)
\end{aligned}
$$

$$
\begin{aligned}
&\Psi^{\mathsf{skolem}}_{\mathcal{R}_{19}}A(c)\ \forall Y.[\overline{R(c,Y)} \vee B(Y)] \equiv_{\mathcal{S}}\\
&\bigvee_{\mathcal{B}\subseteq \mathsf{Const}_\Sigma} \Big(\forall X_{\in\mathcal{B}}.\Big[B(X)\Big]\forall X_{\in\mathcal{B}^c}.\Big[\overline{B(X)}\Big]\mathsf{s}(|\mathcal{B}|\cdot m)\\
&\forall X_{\neq c}.\Big[y_1A(X)\Big(Z(X)\Big(\mathsf{s}(|\mathcal{B}^c|)S(X) \vee (-1)\overline{S(X)}\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\\
&\vee\ x_1\overline{Z(X)}S(X)\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\\
&\vee\ \overline{A(X)}\Big(Z(X)\Big(\mathsf{s}(|\mathcal{B}^c|)S(X) \vee (-1)\overline{S(X)}\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\\
&\vee\ \overline{Z(X)}S(X)\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(X,Y)}\Big]\Big)\Big]\\
&y_1A(c)\forall Y_{\in\mathcal{B}^c}.\Big[\overline{R(c,Y)}\Big]\Big(Z(c)\Big(S(c) \vee (-1)\overline{S(c)}\Big) \vee x_1\overline{Z(c)}S(c)\Big)\Big)
\end{aligned}
$$

$$
\tau^{r_1}_{\mathcal{R}_{19}} = \sum_{k=0}^{m}\binom{m}{k}2^{k\cdot m}\ (y_1(2^{m-k}-1+x_1)+2^{m-k})^{m-1}
$$

$$y_1(z(2^{m-k}-1+x_1)+x_1)$$

▶ Knowledge Base $\mathcal{R}_{20}$

We consider $\mathsf{Pred}_\Sigma = \{A/1, B/1, R/2\}$, and $\mathcal{R}_{21} = (\mathcal{F}, \mathcal{B})$ with

- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1\}$ with $r_1 = (\exists Y.R(X,Y) \wedge B(Y)|A(X))[p_1]$.

The difficulty of $\mathcal{R}_{20}$ does not differ from $\mathcal{R}_{19}$ since

$$\begin{aligned} & (\exists Y.R(X,Y) \wedge A(Y)|A(X))[p_1] \\ \equiv_{\mathcal{S}} \; & (\neg\exists Y.R(X,Y) \wedge A(Y)|A(X))[1-p_1] \\ \equiv_{\mathcal{S}} \; & (\forall Y.\neg R(X,Y) \vee \neg A(Y)|A(X))[1-p_1] \end{aligned}$$

▶ Example: Bird Sanctuary

We consider $\mathcal{R}_{\mathsf{sanct}} = (\mathcal{F}, \mathcal{B})$ [Thimm and Kern-Isberner, 2012] with

- $\mathsf{Pred}_\Sigma = \{\mathsf{StripedSeaEagle}/1, \mathsf{SnoringOstrich}/1, \mathsf{Flies}/1, \mathsf{Pink}/1\}$,
- $\mathcal{F} = \{F_1, \ldots, F_4\}$ with

$$\begin{aligned} F_1 &= \forall X.\neg\mathsf{StripedSeaEagle}(X) \vee \neg\mathsf{SnoringOstrich}(X), \\ F_2 &= \forall X.\mathsf{StripedSeaEagle}(X) \Rightarrow \mathsf{Flies}(X), \\ F_3 &= \forall X.\mathsf{SnoringOstrich} \Rightarrow \neg\mathsf{Flies}(X), \\ F_4 &= \forall X.\mathsf{SnoringOstrich} \Rightarrow \mathsf{Pink}(X), \end{aligned}$$

- $\mathcal{B} = \{r_1, r_2, r_3\}$ with

$$\begin{aligned} r_1 &= (\mathsf{StripedSeaEagle}(X)|\top)[0.999], \\ r_2 &= (\mathsf{SnoringOstrich}(X)|\top)[0.001], \\ r_3 &= (\mathsf{Pink}(X)|\mathsf{StripedSeaEagle}(X))[0.001]. \end{aligned}$$

Typed model counting for $\mathcal{R}_{\mathsf{sanct}}$ can be performed similar to $\mathcal{R}_8$.

▶ Example: Blue Ball

We consider $\mathcal{R}_{\mathsf{ball}} = (\mathcal{F}, \mathcal{B})$ [Thimm and Kern-Isberner, 2012] with

- $\mathsf{Pred}_\Sigma = \{\mathsf{Ball}/1\mathsf{Blue}/1\}$,
- $c \in \mathsf{Const}_\Sigma$,
- $\mathcal{F} = \{F_1, F_2\}$ with
$$F_1 = \forall X.\mathsf{Ball}(X),$$
$$F_2 = \mathsf{Blue}(c),$$
- $\mathcal{B} = \{r_1\}$ with
$$r_1 = (\mathsf{Blue}(X)|\mathsf{Ball}(X))[0.9].$$

Typed model counting for $\mathcal{R}_{\mathsf{ball}}$ can be performed similar to $\mathcal{R}_6$ and $\mathcal{R}_8$.

▶ Example: SYN

We consider $\mathcal{R}_{\mathsf{SYN}} = (\mathcal{F}, \mathcal{B})$ [Finthammer and Beierle, 2012] with

- $\mathsf{Pred}_\Sigma = \{R/1, Q/1\}$,
- $a \in \mathsf{Const}_\Sigma$,
- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1, r_2, r_3\}$ with
$$r_1 = (R(X)|Q(X))[0.7],$$
$$r_2 = (Q(X)|\top)[0.2],$$
$$r_3 = (Q(a)|\top)[0.6].$$

Typed model counting for $\mathcal{R}_{\mathsf{SYN}}$ can be performed similar to $\mathcal{R}_4$.

▶ Example: Asymmetry

We consider $\mathcal{R}_{\mathsf{asym}} = (\mathcal{F}, \mathcal{B})$ [Jaeger, 1994] with

- $\mathsf{Pred}_\Sigma = \{A/1, B/1, C/1\}$,
- $a \in \mathsf{Const}_\Sigma$,
- $\mathcal{F} = \{F_1\}$ with
$$F_1 = \forall X.C(X) \Rightarrow A(X)B(X),$$
- $\mathcal{B} = \{r_1, \ldots, r_4\}$ with
$$\begin{aligned} r_1 &= (C(X)|A(X))[0.1],\\ r_2 &= (C(X)|B(X))[0.9],\\ r_3 &= (A(a)|\top)[0.5],\\ r_4 &= (B(a)|\top)[0.5]. \end{aligned}$$

Typed model counting for $\mathcal{R}_{\mathsf{asym}}$ can be performed similar to $\mathcal{R}_4$ and $\mathcal{R}_8$.

▶ Example: Antarctic Birds

We consider $\mathcal{R}_{\mathsf{arct}} = (\mathcal{F}, \mathcal{B})$ [Jaeger, 1994] with

- $\mathsf{Pred}_\Sigma = \{\mathsf{Bird}/1, \mathsf{FlyingBird}/1, \mathsf{AntarcticBird}/1\}$,
- $\mathsf{opus} \in \mathsf{Const}_\Sigma$,
- $\mathcal{F} = \{F_1, F_2, F_3\}$ with
$$\begin{aligned} F_1 &= \forall X.\mathsf{FlyingBird}(X) \Rightarrow \mathsf{Bird}(X),\\ F_2 &= \forall X.\mathsf{AntarcticBird}(X) \Rightarrow \mathsf{Bird}(X),\\ F_3 &= \mathsf{Bird}(\mathsf{opus}), \end{aligned}$$
- $\mathcal{B} = \{r_1, \ldots, r_4\}$ with
$$\begin{aligned} r_1 &= (\mathsf{FlyingBird}(X)|\mathsf{Bird}(X))[0.95],\\ r_2 &= (\mathsf{AntarcticBird}(X)|\mathsf{Bird}(X))[0.01],\\ r_3 &= (\mathsf{FlyingBird}(X)|\mathsf{AntarcticBird}(X))[0.2],\\ r_4 &= (\mathsf{AntarcticBird}(\mathsf{opus})|\top)[0.9]. \end{aligned}$$

Typed model counting for $\mathcal{R}_{\mathsf{arct}}$ can be performed similar to $\mathcal{R}_4$, $\mathcal{R}_6$ and $\mathcal{R}_8$.

▶ Example: Penguin

We consider $\mathcal{R}_{peng} = (\mathcal{F}, \mathcal{B})$ [Thimm and Kern-Isberner, 2012] with

- $\mathsf{Pred}_\Sigma = \{\mathsf{Bird}/1, \mathsf{Penguin}/1, \mathsf{Flies}/1\}$,
- $\mathsf{opus}, \mathsf{tweety}, \mathsf{brian} \in \mathsf{Const}_\Sigma$,
- $\mathcal{F} = \{F_1, F_2, F_3\}$ with
$$F_1 = \mathsf{Bird}(\mathsf{tweety}),$$
$$F_2 = \mathsf{Bird}(\mathsf{opus}),$$
$$F_3 = \mathsf{Bird}(\mathsf{brian}),$$
- $\mathcal{B} = \{r_1, r_2, r_3\}$ with
$$r_1 = (\mathsf{Penguin}(\mathsf{opus})|\top)[0.9],$$
$$r_2 = (\mathsf{Flies}(X)|\mathsf{Bird}(X))[0.6],$$
$$r_3 = (\mathsf{Flies}(X)|\mathsf{Penguin}(X))[0.01].$$

Typed model counting for $\mathcal{R}_{\mathsf{peng}}$ can be performed similar to $\mathcal{R}_4$ and $\mathcal{R}_6$.

▶ Example: Chirps

We consider $\mathcal{R}_{\mathsf{chirps}} = (\mathcal{F}, \mathcal{B})$ [Thimm et al., 2011] with

- $\mathsf{Pred}_\Sigma = \{\mathsf{Chirps}/1, \mathsf{Bird}/1, \mathsf{Magpie}/1, \mathsf{Moody}/1\}$,
- $\mathsf{tweety} \in \mathsf{Const}_\Sigma$,
- $\mathcal{F} = \{F_1, F_2\}$ with
$$F_1 = \forall X.\mathsf{Magpie}(X) \Rightarrow \mathsf{Bird}(X),$$
$$F_2 = \mathsf{Magpie}(\mathsf{tweety}),$$
- $\mathcal{B} = \{r_1, r_2\}$ with
$$r_1 = (\mathsf{Chirps}(X)|\mathsf{Bird}(X))[0.9],$$
$$r_2 = (\mathsf{Chirps}(X)|\mathsf{Magpie}(X) \wedge \mathsf{Moody}(X))[0.2].$$

Typed model counting for $\mathcal{R}_{\mathsf{chirps}}$ can be performed similar to $\mathcal{R}_6$ and $\mathcal{R}_8$.

▶ Example: Flu

We consider $\mathcal{R}_{\mathsf{flu}} = (\mathcal{F}, \mathcal{B})$ [Finthammer and Beierle, 2012] with

- $\mathsf{Pred}_{\Sigma} = \{\mathsf{Flu}/1, \mathsf{Contact}/2\}$,
- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1, r_2\}$ with
$$r_1 = (\mathsf{Flu}(X)|\top)[0.2],$$
$$r_2 = (\mathsf{Flu}(X)|\mathsf{Contact}(X,Y) \wedge \mathsf{Flu}(Y))[0.4].$$

Typed model counting for $\mathcal{R}_{\mathsf{flu}}$ can be performed similar to $\mathcal{R}_{15}$ using set disjunction.

▶ Example: Flu Extended

We consider $\mathcal{R}_{\mathsf{susc}} = (\mathcal{F}, \mathcal{B})$ [Thimm and Kern-Isberner, 2012] with

- $\mathsf{Pred}_{\Sigma} = \{\mathsf{Flu}/1, \mathsf{Susceptible}/1, \mathsf{Contact}/2\}$,
- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1, r_2, r_3\}$ with
$$r_1 = (\mathsf{Flu}(X)|\top)[0.2],$$
$$r_2 = (\mathsf{Flu}(X)|\mathsf{Contact}(X,Y) \wedge \mathsf{Flu}(Y))[0.4],$$
$$r_3 = (\mathsf{Flu}(X)|\mathsf{Susceptible}(X))[0.3].$$

Typed model counting for $\mathcal{R}_{\mathsf{susc}}$ can be performed similar to $\mathcal{R}_{15}$ using set disjunction.

▶ Example: Cold

We consider $\mathcal{R}_{\mathsf{cold}} = (\mathcal{F}, \mathcal{B})$ [Thimm et al., 2011] with

- $\mathsf{Pred}_{\Sigma} = \{\mathsf{Cold}/1, \mathsf{Susceptible}/1, \mathsf{Contact}/1\}$,
- $\mathcal{F} = \{F_1, F_2\}$ with
$$F_1 = \forall X. \neg \mathsf{Contact}(X, X),$$

$$F_2 = \forall X, Y.\mathsf{Contact}(X,Y) \Rightarrow \mathsf{Contact}(Y,X),$$

- $\mathcal{B} = \{r_1, r_2, r_3\}$ with
$$\begin{aligned} r_1 &= (\mathsf{Cold}(X)|\top)[0.01], \\ r_2 &= (\mathsf{Cold}(X)|\mathsf{Susceptible}(X))[0.1], \\ r_3 &= (\mathsf{Cold}(X)|\mathsf{Contact}(X,Y) \wedge \mathsf{Cold}(Y))[0.6]. \end{aligned}$$

Typed model counting for $\mathcal{R}_{\mathsf{cold}}$ can be performed similar to $\mathcal{R}_8$, $\mathcal{R}_{15}$, and $\mathcal{R}_{16}$ using set disjunction and paired grounding.

▶ Example: Monkeys

We consider $\mathcal{R}_{\mathsf{monk}} = (\mathcal{F}, \mathcal{B})$ [Finthammer and Beierle, 2012] with

- $\mathsf{Pred}_\Sigma = \{\mathsf{Feeds}/2, \mathsf{Hungry}/1\}$,
- $\mathsf{charly} \in \mathsf{Const}_\Sigma$,
- $\mathcal{F} = \{F_1\}$ with
$$F_1 = \forall X.\neg\mathsf{Feeds}(X,X)$$
- $\mathcal{B} = \{r_1, \dots, r_5\}$ with
$$\begin{aligned} r_1 &= (\mathsf{Feeds}(X,Y)|\neg\mathsf{Hungry}(X) \wedge \mathsf{Hungry}(Y))[0.8], \\ r_2 &= (\neg\mathsf{Feeds}(X,Y)|\mathsf{Hungry}(X))[0.999], \\ r_3 &= (\neg\mathsf{Feeds}(X,Y)|\neg\mathsf{Hungry}(X) \wedge \neg\mathsf{Hungry}(Y))[0.9], \\ r_4 &= (\mathsf{Feeds}(X,\mathsf{charly})|\neg\mathsf{Hungry}(X))[0.95]. \end{aligned}$$

Typed model counting for $\mathcal{R}_{\mathsf{monk}}$ can be performed similar to $\mathcal{R}_4$, $\mathcal{R}_8$, and $\mathcal{R}_{15}$ using set disjunction.

▶ Example: Elephants and Keepers

We consider $\mathcal{R}_{\mathsf{eleph}} = (\mathcal{F}, \mathcal{B})$ [Thimm and Kern-Isberner, 2012] with

- $\mathsf{Pred}_\Sigma = \{\mathsf{Elephant}/1, \mathsf{Keeper}/1, \mathsf{Likes}/2\}$,

- $\mathcal{F} = \{F_1, \ldots, F_4\}$ with

$$\begin{aligned} F_1 &= \mathsf{Elephant}(\mathsf{clyde}), \\ F_2 &= \mathsf{Elephant}(\mathsf{giddy}), \\ F_3 &= \mathsf{Keeper}(\mathsf{fred}), \\ F_4 &= \mathsf{Keeper}(\mathsf{dave}), \end{aligned}$$

- $\mathcal{B} = \{r_1, r_2, r_3\}$ with

$$\begin{aligned} r_1 &= (\mathsf{Likes}(X, Y)|\mathsf{Elephant}(X) \wedge \mathsf{Keeper}(Y))[0.6], \\ r_2 &= (\mathsf{Likes}(X, \mathsf{fred})|\mathsf{Elephant}(X) \wedge \mathsf{Keeper}(\mathsf{fred}))[0.4], \\ r_3 &= (\mathsf{Likes}(\mathsf{clyde}, \mathsf{fred})|\mathsf{Elephant}(\mathsf{clyde}) \wedge \mathsf{Keeper}(\mathsf{fred}))[0.7]. \end{aligned}$$

Typed model counting for $\mathcal{R}_{\mathsf{eleph}}$ can be performed similar to $\mathcal{R}_4$, $\mathcal{R}_6$, and $\mathcal{R}_{15}$ using set disjunction.

▶ Example: Burglary

We consider $\mathcal{R}_{\mathsf{burg}} = (\mathcal{F}, \mathcal{B})$ [Beierle et al., 2010] with

- $\mathsf{Pred}_\Sigma = \{\mathsf{Alarm}/1, \mathsf{Burglary}/1, \mathsf{Tornado}/1, \mathsf{LivesIn}/2, \mathsf{Neighborhood}/2\}$,
- $\mathsf{carl}, \mathsf{james}, \mathsf{austin}, \mathsf{yorkshire}, \mathsf{bad}, \mathsf{average}, \mathsf{good} \in \mathsf{Const}_\Sigma$,
- $\mathcal{F} = \{F_1, \ldots, F_8\}$ with

$$\begin{aligned} F_1 &= \forall X, Y, Z, Y \neq Z. \mathsf{Neighborhood}(X, Y) \Rightarrow \mathsf{Neighborhood}(X, Z), \\ F_2 &= \forall X, Y, Z, Y \neq Z. \mathsf{LivesIn}(X, Y) \Rightarrow \neg \mathsf{LivesIn}(X, Z), \\ F_3 &= \mathsf{LivesIn}(\mathsf{james}, \mathsf{yorkshire}), \\ F_4 &= \mathsf{LivesIn}(\mathsf{carl}, \mathsf{austin}), \\ F_5 &= \mathsf{Burglary}(\mathsf{james}), \\ F_6 &= \mathsf{Tornado}(\mathsf{austin}), \\ F_7 &= \mathsf{Neighborhood}(\mathsf{james}, \mathsf{average}), \\ F_8 &= \mathsf{Neighborhood}(\mathsf{carl}, \mathsf{good}), \end{aligned}$$

- $\mathcal{B} = \{r_1, \ldots, r_5\}$ with

$$\begin{aligned} r_1 &= (\mathsf{Alarm}(X)|\mathsf{Burglary}(X))[0.9], \\ r_2 &= (\mathsf{Alarm}(X)|\mathsf{LivesIn}(X, Y) \wedge \mathsf{Tornado}(Y))[0.9], \\ r_3 &= (\mathsf{Burglary}(X)|\mathsf{Neighborhood}(X, \mathsf{bad}))[0.6], \\ r_4 &= (\mathsf{Burglary}(X)|\mathsf{Neighborhood}(X, \mathsf{average}))[0.4], \end{aligned}$$

$$r_5 = (\mathsf{Burglary}(X)|\mathsf{Neighborhood}(X, \mathsf{good}))[0.3].$$

Typed model counting for $\mathcal{R}_{\mathsf{burg}}$ can be performed similar to $\mathcal{R}_4$, $\mathcal{R}_6$, $\mathcal{R}_8$, and $\mathcal{R}_{15}$ using set disjunction.

▶ Example: Lottery

We consider $\mathcal{R}_{\mathsf{lottery}} = (\mathcal{F}, \mathcal{B})$ [Delgrande, 1998] with

- $\mathsf{Pred}_\Sigma = \{\mathsf{Winner}/1\}$,
- $\mathcal{F} = \emptyset$,
- $\mathcal{B} = \{r_1, r_2\}$ with

$$\begin{aligned} r_1 &= (\mathsf{Winner}(X)|\top)[0.001], \\ r_2 &= (\exists X.\mathsf{Winner}(X)|\top)[0.7]. \end{aligned}$$

Typed model counting for $\mathcal{R}_{\mathsf{lottery}}$ can be performed using skolemization.

▶ Example: DL-Cold

We consider $\mathcal{R}_{dlcold} = (\mathcal{F}, \mathcal{B})$ [Thimm et al., 2011] with

- $\mathsf{Pred}_\Sigma = \{\mathsf{Cold}/1, \mathsf{Susceptible}/1, \mathsf{Contact}/2\}$,
- $\mathcal{F} = \{F_1\}$ with

$$F_1 = \forall X.\exists Y.\mathsf{Contact}(X, Y),$$

- $\mathcal{B} = \{r_1, r_2, r_3\}$ with

$$\begin{aligned} r_1 &= (\mathsf{Cold}(X)|\top)[0.01], \\ r_2 &= (\mathsf{Cold}(X)|\mathsf{Susceptible}(X))[0.1], \\ r_3 &= (\mathsf{Cold}(X)|\exists Y.\mathsf{Contact}(X, Y) \land \mathsf{Cold}(Y))[0.6]. \end{aligned}$$

Typed model counting for $\mathcal{R}_{\mathsf{dlcold}}$ can be performed similar to $\mathcal{R}_8$, $\mathcal{R}_{15}$, and $\mathcal{R}_{19}$ using set disjunction and skolemization.

▶ Example: Antibiotics

We consider $\mathcal{R}_{\mathsf{biot}} = (\mathcal{F}, \mathcal{B})$ [Peñaloza and Potyka, 2016] with

- $\mathsf{Pred}_\Sigma = \{\mathsf{Antibiotic}/1, \mathsf{BacterialInfection}/1, \mathsf{AntibioticResistant}/1, \mathsf{HeavyAntibioticUse}/1, \mathsf{ViralInfection}/1, \mathsf{Infection}/1, \mathsf{SF}/2, \mathsf{ST}/2\}$, with SF = suffers from and ST = successful treatment,
- $\mathcal{F} = \{F_1, \dots, F_5\}$ with

$$\begin{aligned} F_1 = \forall X.(&\exists Y.\mathsf{SF}(X,Y) \land \mathsf{BacterialInfection}(Y) \land \neg\mathsf{AntibioticResistant}(Y) \\ &\land \neg\mathsf{HeavyAntibioticUse}(Y)) \Rightarrow \exists Y.\mathsf{ST}(X,Y) \land \mathsf{Antibiotic}(Y), \\ F_2 = \forall X.(&\exists Y.\mathsf{SF}(X,Y) \land \mathsf{BacterialInfection}(Y) \land \mathsf{AntibioticResistant}(Y)) \\ &\Rightarrow \neg\exists Y.\mathsf{ST}(X,Y) \land \mathsf{Antibiotic}(Y), \\ F_3 = \forall X.(&\exists Y.\mathsf{SF}(X,Y) \land \mathsf{ViralInfection}(Y)) \Rightarrow \neg\exists\mathsf{ST}(X,Y) \land \mathsf{Antibiotic}(Y), \\ F_4 = \forall X.&\mathsf{BacterialInfection}(X) \Rightarrow \mathsf{Infection}(X), \\ F_5 = \forall X.&\mathsf{ViralInfection}(X) \Rightarrow \mathsf{Infection}(X), \end{aligned}$$

- $\mathcal{B} = \{r_1, r_2\}$ with

$$\begin{aligned} r_1 &= (\mathsf{AntibioticResistant}(X) | \top)[0.05], \\ r_2 &= (\mathsf{AntibioticResistant}(X) | \mathsf{HeavyAntibioticUse}(X))[0.8]. \end{aligned}$$

Typed model counting for $\mathcal{R}_{\mathsf{biot}}$ can be performed similar to $\mathcal{R}_8$, $\mathcal{R}_{15}$, and $\mathcal{R}_{18}$ using set disjunction and skolemization.

▶ Example: High Blood Pressure

We consider $\mathcal{R}_{\mathsf{blood}} = (\mathcal{F}, \mathcal{B})$ [Lukasiewicz, 2008] with

- $\mathsf{Pred}_\Sigma = \{\mathsf{HeartPatient}/1, \mathsf{PaceMakerPatient}/1, \mathsf{HasSymptom}/2, \mathsf{Male}/1, \mathsf{Arrhythmia}/1, \mathsf{HighBloodPressure}/1, \mathsf{HasHealthInsurance}/2, \mathsf{Private}/1, \mathsf{ChestPain}/1, \mathsf{BreathingDifficulties}/1, \mathsf{HasIllnessStatus}/2, \mathsf{Final}/1, \}$,
- $\mathsf{tom}, \mathsf{john}, \mathsf{maria} \in \mathsf{Const}_\Sigma$,
- $\mathcal{F} = \{F_1, \dots, F_6\}$ with

$$\begin{aligned} F_1 &= \forall X.\mathsf{PaceMakerPatient}(X) \Rightarrow \mathsf{HeartPatient}(X), \\ F_2 &= \forall X.(\exists Y.\mathsf{HasSymptom}(X,Y)) \Rightarrow \mathsf{HeartPatient}(X), \\ F_3 &= \mathsf{HeartPatient}(\mathsf{tom}), \end{aligned}$$

$F_4 = \mathsf{PaceMakerPatient}(\mathsf{john}) \wedge \mathsf{Male}(\mathsf{john}),$
$F_5 = \mathsf{PaceMakerPatient}(\mathsf{maria}) \wedge \neg\mathsf{Male}(\mathsf{maria}),$
$F_6 = \exists Y.\mathsf{HasSymptom}(\mathsf{john}, Y) \wedge \mathsf{Arrhythmia}(Y),$

- $\mathcal{B} = \{r_1, \dots, r_{11}\}$ with
  $r_1 = (\mathsf{HighBloodPressure}(X)|\mathsf{HeartPatient}(X))[0.7],$
  $r_2 = (\neg\mathsf{HighBloodPressure}(X)|\mathsf{PaceMakerPatient}(X))[0.6],$
  $r_3 = (\mathsf{Male}(X)|\mathsf{PaceMakerPatient}(X))[0.4],$
  $r_4 = (\exists Y.\mathsf{HasHealthInsurance}(X, Y)\mathsf{Private}(Y)|\mathsf{HeartPatient}(X))[0.9],$
  $r_5 = (\exists Y.\mathsf{HasSymptom}(X, Y)\mathsf{Arrhythmia}(Y)|\mathsf{PaceMakerPatient}(X))[0.98],$
  $r_6 = (\exists Y.\mathsf{HasSymptom}(X, Y)\mathsf{ChestPain}(Y)|\mathsf{PaceMakerPatient}(X))[0.9],$
  $r_7 = (\exists Y.\mathsf{HasSymptom}(X, Y)\mathsf{BreathingDifficulties}(Y)|\mathsf{PaceMakerPatient}(X))[0.6],$
  $r_8 = (\mathsf{PaceMakerPatient}(\mathsf{tom})|\top)[0.8],$
  $r_9 = (\exists Y.\mathsf{HasSymptom}(\mathsf{maria}, Y) \wedge \mathsf{BreathingDifficulties}(Y)|\top)[0.6],$
  $r_{10} = (\exists Y.\mathsf{HasSymptom}(\mathsf{maria}, Y) \wedge \mathsf{ChestPain}(Y)|\top)[0.9],$
  $r_{11} = (\exists Y.\mathsf{HasIllnessStatus}(\mathsf{maria}, Y) \wedge \mathsf{Final}(Y)|\top)[0.5].$

Typed model counting for $\mathcal{R}_{\mathsf{blood}}$ can be performed similar to $\mathcal{R}_6$, $\mathcal{R}_8$, $\mathcal{R}_{15}$, and $\mathcal{R}_{18}$ using set disjunction and skolemization.

A.3 Own Publications

This is a complete list of peer-reviewed publications that I contributed to during my time as a PhD student until the day my thesis was submitted:

p01 **Gabriele Kern-Isberner, Marco Wilhelm, and Christoph Beierle (2014):** Probabilistic Knowledge Representation Using Gröbner Basis Theory. *International Symposium on Artificial Intelligence and Mathematics, ISAIM 2014, Fort Lauderdale, FL, USA, January 6-8, 2014*

p02 **Gabriele Kern-Isberner, Marco Wilhelm, and Christoph Beierle (2014):** A Novel Methodology for Processing Probabilistic Knowledge Bases Under Maximum Entropy. *Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference, FLAIRS 2014, Pensacola Beach, Florida, USA, May 21-23, 2014. AAAI Press*

p03 **Marco Wilhelm, Gabriele Kern-Isberner, and Andreas Ecke (2016):** Propositional Probabilistic Reasoning at Maximum Entropy Modulo Theories.

Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2016, Key Largo, Florida, USA, May 16-18, 2016. AAAI Press

p04 **Gabriele Kern-Isberner, Marco Wilhelm, and Christoph Beierle (2017):** Probabilistic knowledge representation using the principle of maximum entropy and Gröbner basis theory. *Ann. Math. Artif. Intell., volume 79(1-3)*

p05 **Marco Wilhelm and Gabriele Kern-Isberner (2017):** Typed Model Counting and Its Application to Probabilistic Conditional Reasoning at Maximum Entropy. *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2017, Marco Island, Florida, USA, May 22-24, 2017. AAAI Press*

p06 **Marco Wilhelm, Christian Eichhorn, Richard Niland, and Gabriele Kern-Isberner (2017):** A Semantics for Conditionals with Default Negation. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 14th European Conference, ECSQARU 2017, Lugano, Switzerland, July 10-14, 2017, Proceedings. Lecture Notes in Computer Science, volume 10369. Springer*

p07 **Marco Wilhelm, Marc Finthammer, Gabriele Kern-Isberner, and Christoph Beierle (2017):** First-Order Typed Model Counting for Probabilistic Conditional Reasoning at Maximum Entropy. *Scalable Uncertainty Management - 11th International Conference, SUM 2017, Granada, Spain, October 4-6, 2017, Proceedings. Lecture Notes in Computer Science, volume 10564. Springer*

p08 **Marco Wilhelm, Gabriele Kern-Isberner, and Andreas Ecke (2017):** Basic Independence Results for Maximum Entropy Reasoning Based on Relational Conditionals. *GCAI 2017, 3rd Global Conference on Artificial Intelligence, Miami, FL, USA, 18-22 October 2017, EPiC Series in Computing, volume 50. EasyChair*

p09 **Gabriele Kern-Isberner, Marco Wilhelm, and Christoph Beierle (2018):** Drawing Inferences Under Maximum Entropy From Relational Probabilistic Knowledge Using Group Theory. *Infinite Group Theory: From the Past to the Future, chapter 9. World Scientific*

p10 **Marco Wilhelm, Gabriele Kern-Isberner, Marc Finthammer, and Christoph Beierle (2018):** A Generalized Iterative Scaling Algorithm for

Maximum Entropy Model Computations Respecting Probabilistic Independencies. *Foundations of Information and Knowledge Systems - 10th International Symposium, FoIKS 2018, Budapest, Hungary, May 14-18, 2018, Proceedings. Lecture Notes in Computer Science, volume 10833. Springer*

p11 **Jonas Philipp Haldimann, Marco Wilhelm, and Gabriele Kern-Isberner (2018):** Evaluating Reactive ASP by Formal Belief Revision. *Workshop on Hybrid Reasoning and Learning (HRL 2018) at KR 2018*

p12 **Marco Wilhelm, Gabriele Kern-Isberner, Andreas Ecke, and Franz Baader (2019):** Counting Strategies for the Probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$ Under the Principle of Maximum Entropy. *Logics in Artificial Intelligence - 16th European Conference, JELIA 2019, Rende, Italy, May 7-11, 2019, Proceedings. Lecture Notes in Computer Science, volume 11468. Springer*

p13 **Marco Wilhelm, Gabriele Kern-Isberner, Marc Finthammer, and Christoph Beierle (2019):** Integrating Typed Model Counting into First-Order Maximum Entropy Computations and the Connection to Markov Logic Networks. *Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference, Sarasota, Florida, USA, May 19-22 2019. AAAI Press*

p14 **Marco Wilhelm and Gabriele Kern-Isberner (2019):** Maximum Entropy Calculations for the Probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$. *Description Logic, Theory Combination, and All That - Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday. Lecture Notes in Computer Science, volume 11560. Springer*

p15 **Franz Baader, Andreas Ecke, Gabriele Kern-Isberner, and Marco Wilhelm (2019):** The Complexity of the Consistency Problem in the Probabilistic Description Logic $\mathcal{ALC}^{\mathsf{ME}}$. *Frontiers of Combining Systems. Springer*

p16 **Marco Wilhelm and Gabriele Kern-Isberner (2020):** Context-Based Inferences from Probabilistic Conditionals with Default Negation at Maximum Entropy. *Proceedings of the Thirty-Third International Florida Artificial Intelligence Research Society Conference, Originally to be held in North Miami Beach, Florida, USA, May 17-20, 2020. AAAI Press*

p17 **Marco Wilhelm and Gabriele Kern-Isberner (2020):** Probabilistic Belief Fusion at Maximum Entropy by First-Order Embedding: *NMR 2020 - 18th International Workshop on Non-Monotonic Reasoning at KR 2020*

p18 **Marco Wilhelm and Gabriele Kern-Isberner (2021):** Focused Inference and System P. *Thirty-Fifth AAAI Conference on Artificial Intelligence,* *AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021. AAAI Press*

p19 **Marco Wilhelm and Gabriele Kern-Isberner (2021):** Predicting Human Responses to Syllogism Tasks Following the Principle of Maximum Entropy (Abstract). *Entropy 2021: The Scientific Tool of the 21st Century. MDPI*

p20 **Marco Wilhelm, Diana Howey, Gabriele Kern-Isberner, Kai Sauerwald, and Christoph Beierle (2021):** A Brief Introduction Into Activation-Based Conditional Inference. *Proceedings of the 7th Workshop on Formal and Cognitive Reasoning co-located with the 44th German Conference on Artificial Intelligence (KI 2021), September 28, 2021. CEUR Workshop Proceedings, volume 2961. CEUR-WS.org*

p21 **Marco Wilhelm, Diana Howey, Gabriele Kern-Isberner, Kai Sauerwald, and Christoph Beierle (2021):** Conditional Inference and Activation of Knowledge Entities in ACT-R. *CoRR, volume abs/2110.15214*

p22 **Marco Wilhelm, Diana Howey, Gabriele Kern-Isberner, Kai Sauerwald, and Christoph Beierle (2022):** Integrating Cognitive Principles From ACT-R Into Probabilistic Conditional Reasoning by Taking the Example of Maximum Entropy Reasoning. *Proceedings of the Thirty-Fifth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2022, Hutchinson Island, Jensen Beach, Florida, USA, May 15-18, 2022*

p23 **Pascal Kaiser, Andre Thevapalan, Moritz Roidl, Gabriele Kern-Isberner, and Marco Wilhelm (2023):** Towards Finding Optimal Solutions For Constrained Warehouse Layouts Using Answer Set Programming. *Proceedings of the Conference on Production Systems and Logistics: CPSL 2023 - 1. publish-Ing., Hannover*

p24 **Marco Wilhelm, Diana Howey, Gabriele Kern-Isberner, Kai Sauerwald, and Christoph Beierle (2023):** Activation-based Conditional Inference. *ifColog, FLAP, volume 10(2)*

p25 **Andre Thevapalan, Marco Wilhelm, Gabriele Kern-Isberner, Pascal Kaiser, and Moritz Roidl (2023):** An Interactive Modelling Environment for Designing Warehouse Layouts Based on ASP. *Proceedings of the Thirty-Sixth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2023, Clearwater Beach, FL, USA, May 14-17, 2023. AAAI Press* p26 **Marco Wilhelm, Andre Thevapalan, and Gabriele Kern-Isberner (2023):** Prioritizing Answer Sets Based on Conditional Expert Knowledge. *Proceedings of the Thirty-Sixth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2023, Clearwater Beach, FL, USA, May 14-17, 2023. AAAI Press*

p27 **Marco Wilhelm (2023):** Integrating Linear Arithmetic Constraints Into Conditional Maximum Entropy Reasoning. *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, KR 2023, Rhodes, Greece, September 2-8, 2023*

p28 **Marco Wilhelm, Meliha Sezgin, Gabriele Kern-Isberner, Jonas Haldimann, Christoph Beierle, and Jesse Heyninck (2023):** Splitting Techniques for Conditional Belief Bases in the Context of c-Representations. *Logics in Artificial Intelligence - 18th European Conference, JELIA 2023, Dresden, Germany, September 20-22, 2023, Proceedings. Lecture Notes in Computer Science, volume 14281. Springer*

p29 **Marco Wilhelm, Diana Howey, Gabriele Kern-Isberner, Kai Sauerwald, and Christoph Beierle (2023):** Activation-Based Conditional Inference (Extended Abstract). *2nd Workshop on Challenges and Adequacy Conditions for Logics in the New Age of Artificial Intelligence (ACLAI)*

p30 **Marco Wilhelm and Gabriele Kern-Isberner (2024):** Decomposing Constraint Networks for Calculating c-Representations. *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada. AAAI Press*

p31 **Marco Wilhelm, Gabriele Kern-Isberner, and Christoph Beierle (2024):** Core c-Representations and c-Core Closure for Conditional Belief Bases. *Foundations of Information and Knowledge Systems - 13th International Symposium, FoIKS 2024, Sheffield, UK, April 8-11, 2024, Proceedings. Lecture Notes in Computer Science, volume 14589. Springer*

p32 **Marco Wilhelm, Lars-Phillip Spiegel, and Gabriele Kern-Isberner (2024):** Intrinsic Prioritization in Answer Set Programming Based on an Adapted Notion of Tolerance. *Proceedings of the Thirty-Seventh International Florida Artificial Intelligence Research Society Conference, FLAIRS 2024, Miramar Beach, FL, USA, May 19-21, 2024. AAAI Press*

Bibliography

Adams, E. W. (1975). *The Logic of Conditionals. An Application of Probability to Deductive Logic*, volume 86 of *Synthese Library*. Springer.

Alchourrón, C. E., Gärdenfors, P., and Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510—530.

Alpaydin, E. (2020). *Introduction to Machine Learning*. MIT Press, 4 edition.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060.

Apsel, U. and Brafman, R. I. (2011). Extended lifted inference with joint formulas. In Cozman, F. G. and Pfeffer, A., editors, *UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011*, pages 11–18. AUAI Press.

Baader, F. and Bortoli, F. D. (2023). On the abstract expressive power of description logics with concrete domains. In Kutz, O., Lutz, C., and Ozaki, A., editors, *Proceedings of the 36th International Workshop on Description Logics (DL 2023) co-located with the 20th International Conference on Principles of Knowledge Representation and Reasoning and the 21st International Workshop on Non-Monotonic Reasoning (KR 2023 and NMR 2023)., Rhodes, Greece, September 2-4, 2023*, volume 3515 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Baader, F. and Bortoli, F. D. (2024). The abstract expressive power of first-order and description logics with concrete domains. In Hong, J. and Park, J. W., editors, *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC 2024, Avila, Spain, April 8-12, 2024*, pages 754–761. ACM.

Baader, F., Brandt, S., and Lutz, C. (2005). Pushing the EL envelope. In Kaelbling, L. P. and Saffiotti, A., editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pages 364–369. Professional Book Center.

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors (2007). *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2 edition.

Baader, F. and Ecke, A. (2017). Extending the description logic ALC with more expressive cardinality constraints on concepts. In Benzmüller, C., Lisetti, C. L., and Theobald, M., editors, *GCAI 2017, 3rd Global Conference on Artificial Intelligence, Miami, FL, USA, 18-22 October 2017*, volume 50 of *EPiC Series in Computing*, pages 6–19. EasyChair.

Baader, F., Ecke, A., Kern-Isberner, G., and Wilhelm, M. (2019). The complexity of the consistency problem in the probabilistic description logic $\mathcal{ALC}^{\mathsf{ME}}$. In Herzig, A. and Popescu, A., editors, *Frontiers of Combining Systems*, pages 167–184. Springer.

Baader, F. and Hanschke, P. (1991). A scheme for integrating concrete domains into concept languages. In Mylopoulos, J. and Reiter, R., editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991*, pages 452–457. Morgan Kaufmann.

Baader, F., Horrocks, I., Lutz, C., and Sattler, U. (2017). *An Introduction to Description Logic.* Cambridge University Press.

Baader, F., Horrocks, I., and Sattler, U. (2008a). Description logics. In van Harmelen, F., Lifschitz, V., and Porter, B., editors, *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 135–179. Elsevier.

Baader, F., Lutz, C., and Brandt, S. (2008b). Pushing the EL envelope further. In Clark, K. and Patel-Schneider, P. F., editors, *Proceedings of the Fourth OWLED Workshop on OWL: Experiences and Directions, Washington, DC, USA, 1-2 April 2008*, volume 496 of *CEUR Workshop Proceedings.* CEUR-WS.org.

Baader, F. and Sattler, U. (2001). An overview of tableau algorithms for description logics. *Studia Logica: An International Journal for Symbolic Logic*, 69(1):5–40.

Bacchus, F., Grove, A. J., Halpern, J. Y., and Koller, D. (1996). From statistical knowledge bases to degrees of belief. *Artif. Intell.*, 87(1-2):75–143.

Baier, L., Jöhren, F., and Seebacher, S. (2019). Challenges in the deployment and operation of machine learning in practice. In vom Brocke, J., Gregor, S., and Müller, O., editors, *27th European Conference on Information Systems - Information Systems for a Sharing Society, ECIS 2019, Stockholm and Uppsala, Sweden, June 8-14, 2019.*

Barnett, O. and Paris, J. B. (2008). Maximum entropy inference with quantified knowledge. *Log. J. IGPL*, 16(1):85–98.

Barrett, C. W., Sebastiani, R., Seshia, S. A., and Tinelli, C. (2021). Satisfiability modulo theories. In Biere, A., Heule, M., van Maaren, H., and Walsh, T., editors, *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, pages 1267–1329. IOS Press.

Bartlett, M. S. (1936). Statistical probability. *Journal of the American Statistical Association*, 31(195):553–555.

Beame, P., Van den Broeck, G., Gribkoff, E., and Suciu, D. (2015). Symmetric weighted first-order model counting. In Milo, T. and Calvanese, D., editors, *Proceedings of the 34th ACM Symposium on Principles of Database Systems, PODS 2015, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 313–328. ACM.

Becker, T. and Weispfenning, B. V. (1993). *Gröbner bases: A computational approach to commutative algebra.* Graduate Texts in Mathematics: Readings in mathematics. Bd. 141. Springer.

Beedkar, K., Del Corro, L., and Gemulla, R. (2013). Fully parallel inference in markov logic networks. In Markl, V., Saake, G., Sattler, K., Hackenbroich, G., Mitschang, B., Härder, T., and Köppen, V., editors, *Datenbanksysteme für Business, Technologie und Web (BTW), 15. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme" (DBIS), 11.-15.3.2013 in Magdeburg, Germany. Proceedings*, volume P-214 of *LNI*, pages 205–224. GI.

Beierle, C., Finthammer, M., Kern-Isberner, G., and Thimm, M. (2010). Automated reasoning for relational probabilistic knowledge representation. In Giesl, J. and Hähnle, R., editors, *Automated Reasoning, 5th International Joint Conference, IJCAR 2010, Edinburgh, UK, July 16-19, 2010. Proceedings*, volume 6173 of *Lecture Notes in Computer Science*, pages 218–224. Springer.

Beierle, C. and Kern-Isberner, G. (2014). *Methoden wissensbasierter Systeme - Grundlagen, Algorithmen, Anwendungen, 5. Auflage.* Computational Intelligence. SpringerVieweg.

Beierle, C., Kuche, S., Finthammer, M., and Kern-Isberner, G. (2015). A software system for the computation, visualization, and comparison of conditional structures for relational probabilistic knowledge bases. In Russell, I. and Eberle, W., editors, *Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2015, Hollywood, Florida, USA, May 18-20, 2015*, pages 558–563. AAAI Press.

Berger, A. L., Della Pietra, S. A., and Della Pietra, V. J. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Bergmann, M., Moor, J., and Nelson, J. (2013). *The Logic Book.* McGraw-Hill Education, 6 edition.

Bertsekas, D. P. (2014). *Constrained Optimization and Lagrange Multiplier Methods.* Academic Press.

Beziau, J.-Y. (2003). Bivalence, excluded middle and non contradiction. In Behounek, L., editor, *The Logica Yearbook 2003*, pages 73–84. Academy of Sciences.

Bolton, R. and Hand, D. (2002). Statistical fraud detection: A review. *Statistical Science*, 17(3):235–255.

Boole, G. (1854). *An Investigation of the Laws of Thought.* Walton and Maberly.

Boolos, G. S., Burgess, J. P., and Jeffrey, R. C. (2007). *Computability and Logic.* Cambridge University Press, 5 edition.

Boyd, S. and Vandenberghe, L. (2009). *Convex Optimization.* Cambridge University Press, 7 edition.

Brachman, R. J. and Levesque, H. J. (2004). *Knowledge Representation and Reasoning.* Elsevier.

Brewka, G. (1991). *Nonmonotonic reasoning - Logical foundations of commonsense*, volume 12 of *Cambridge tracts in theoretical computer science.* Cambridge University Press.

Calabrese, P. G. (1991). Deduction and inference using conditional logic and probability. In Goodman, I. R., Gupta, M. M., Nguyen, H. T., and Rogers, G. S., editors, *Conditional Logic in Expert Systems*, pages 71–100. Elsevier.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., and Rosati, R. (2005). DL-Lite: tractable description logics for ontologies. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI2005)*, pages 602–607.

Calvanese, D., De Giacomo, G., Nardi, D., and Lenzerini, M. (2001). Reasoning in expressive description logics. In *Handbook of Automated Reasoning*, pages 1581–1634. Elsevier.

Chavira, M. and Darwiche, A. (2008). On probabilistic inference by weighted model counting. *Artif. Intell.*, 172(6-7):772–799.

Chong, E. K. P. and Zak, S. H. (2013). Gradient methods. In *An Introduction to Optimization*, pages 131–160. Wiley, 4 edition.

Christensen, R. (1997). *Log-Linear Models and Logistic Regression.* Springer.

Clark, K. L. (1977). Negation as failure. In Gallaire, H. and Minker, J., editors, *Logic and Data Bases, Symposium on Logic and Data Bases, Centre d'études et de recherches de Toulouse, France, 1977*, Advances in Data Base Theory, pages 293–322, New York. Plemum Press.

Covington, M. A., Nute, D., and Vellino, A. (1988). *Prolog Programming in Depth.* Scott, Foresman & Co.

Cox, R. T. (1946). Probability, frequency and reasonable expectation. *American Journal of Physics*, 14(1):1–13.

Csiszar, I. (1989). A Geometric Interpretation of Darroch and Ratcliff's Generalized Iterative Scaling. *The Annals of Statistics*, 17(3):1409 – 1413.

Cummins, D. D., Lubart, T., Alksnis, O., and Rist, R. (1991). Conditional reasoning and causation. *Memory & Cognition*, 19(3):274–282.

Dale, A. I. and Laplace, P.-S. (1995). *Pierre-Simon Laplace: Philosophical Essay on Probabilities.* Springer.

Darroch, J. N. and Ratcliff, D. (1972). Generalized Iterative Scaling for Log-Linear Models. *The Annals of Mathematical Statistics*, 43(5):1470 – 1480.

Darwiche, A. (2001a). Decomposable negation normal form. *J. ACM*, 48(4):608–647.

Darwiche, A. (2001b). On the tractable counting of theory models and its application to truth maintenance and belief revision. *J. Appl. Non Class. Logics*, 11(1-2):11–34.

Darwiche, A. and Marquis, P. (2002). A knowledge compilation map. *J. Artif. Intell. Res.*, 17:229–264.

Darwiche, A. and Pearl, J. (1997). On the logic of iterated belief revision. *Artificial Intelligence*, 89(1):1–29.

de Finetti, B. (2017). *Theory of Probability: A Critical Introductory Treatment.* John Wiley & Sons.

de Laplace, P. S. (1820). *Théorie Analytique des Probabilités.* Courcier.

De Raedt, L., Kersting, K., Natarajan, S., and Poole, D. (2016). *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation.* Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

De Raedt, L. and Kimmig, A. (2015). Probabilistic (logic) programming concepts. *Mach. Learn.*, 100(1):5–47.

De Raedt, L., Kimmig, A., and Toivonen, H. (2007). Problog: A probabilistic prolog and its application in link discovery. In Veloso, M. M., editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2462–2467.

de Salvo Braz, R., Amir, E., and Roth, D. (2008). A survey of first-order probabilistic models. In Holmes, D. E. and Jain, L. C., editors, *Innovations in Bayesian Networks: Theory and Applications*, volume 156 of *Studies in Computational Intelligence*, pages 289–317. Springer.

Delgrande, J. P. (1998). On first-order conditional logics. *Artif. Intell.*, 105(1-2):105–137.

Della Pietra, S., Della Pietra, V., and Lafferty, J. (1997). Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393.

Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics*, 38(2):325–339.

Dilkas, P. and Belle, V. (2023). Synthesising recursive functions for first-order model counting: Challenges, progress, and conjectures. In Marquis, P., Son, T. C., and Kern-Isberner, G., editors, *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, KR 2023, Rhodes, Greece, September 2-8, 2023*, pages 198–207.

Dubois, D., Lang, J., and Prade, H. (1991). A brief overview of possibilistic logic. In Kruse, R. and Siegel, P., editors, *Symbolic and Quantitative Approaches to Uncertainty.* Springer.

Dubois, D. and Prade, H. (1988). *Possibility Theory: An Approach To Computerized Processing Of Uncertainty.* Springer.

Dugdale, J. S. (1996). *Entropy and its Physical Meaning.* Taylor and Francis, 2 edition.

Ebbinghaus, H.-D., Flum, J., and Thomas, W. (2021). *Mathematical Logic.* Springer, 3 edition.

Finthammer, M. (2012). An iterative scaling algorithm for maximum entropy reasoning in relational probabilistic conditional logic. In Hüllermeier, E., Link, S., Fober, T., and Seeger, B., editors, *Scalable Uncertainty Management - 6th International Conference, SUM 2012, Marburg, Germany, September 17-19, 2012. Proceedings*, volume 7520 of *Lecture Notes in Computer Science*, pages 351–364. Springer.

Finthammer, M. (2017). *Concepts and algorithms for computing maximum entropy distributions for knowledge bases with relational probabilistic conditionals.* PhD thesis, University of Hagen, Germany.

Finthammer, M. and Beierle, C. (2012). Using equivalences of worlds for aggregation semantics of relational conditionals. In Glimm, B. and Krüger, A., editors, *KI 2012: Advances in Artificial Intelligence - 35th Annual German Conference on AI, Saarbrücken, Germany, September 24-27, 2012. Proceedings*, volume 7526 of *Lecture Notes in Computer Science*, pages 49–60. Springer.

Finthammer, M. and Beierle, C. (2014). A two-level approach to maximum entropy model computation for relational probabilistic logic based on weighted conditional impacts. In Straccia, U. and Calì, A., editors, *Scalable Uncertainty Management - 8th International Conference, SUM 2014, Oxford, UK, September 15-17, 2014. Proceedings*, volume 8720 of *Lecture Notes in Computer Science*, pages 162–175. Springer.

Finthammer, M. and Thimm, M. (2012). An integrated development environment for probabilistic relational reasoning. *Log. J. IGPL*, 20(5):831–871.

Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222(594-604):309–368.

Fisseler, J. (2010). *Learning and Modeling with Probabilistic Conditional Logic.* IOS Press.

Fisseler, J. (2012). First-order probabilistic conditional logic and maximum entropy. *Log. J. IGPL*, 20(5):796–830.

Fisseler, J., Kern-Isberner, G., Beierle, C., Koch, A., and Müller, C. (2007). Algebraic knowledge discovery using haskell. In Hanus, M., editor, *Practical Aspects of Declarative Languages, 9th International Symposium, PADL 2007, Nice, France, January 14-15, 2007*, volume 4354 of *Lecture Notes in Computer Science*, pages 80–93. Springer.

Frigg, R. and Werndl, C. (2011). Entropy - A guide for the perplexed. In Beisbart, C. and Hartmann, S., editors, *Probabilities in Physics.* Oxford University Press.

Gabbay, D. (1994). Classical vs non-classical logic. In Gabbay, D. M., Hogger, C. J., and Robinson, J. A., editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, volume 2*. Oxford University Press.

Gaggl, S., Rudolph, S., and Schweizer, L. (2016). Fixed-domain reasoning for description logics. In *Proceedings of the Twenty-Second European Conference on Artificial Intelligence*, ECAI'16, pages 819–827. IOS Press.

Gärdenfors, P. (1988). *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press.

Gärdenfors, P. and Rott, H. (1995). Belief revision. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4, pages 35–132. Oxford University Press.

Gelfond, M. (2008). Answer sets. In van Harmelen, F., Lifschitz, V., and Porter, B. W., editors, *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 285–316. Elsevier.

Gelfond, M. and Lifschitz, V. (1988). The stable model semantics for logic programming. In Kowalski, R. A. and Bowen, K. A., editors, *Logic Programming, Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, USA, August 15-19, 1988 (2 Volumes)*, pages 1070–1080. MIT Press.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman and Hall, 3 edition.

Genesereth, M. and Kao, E. J. (2016). *Introduction to Logic*. Morgan & Claypool Publishers, 3 edition.

Getoor, L. and Taskar, B. (2007). *Introduction to Statistical Relational Learning*. MIT Press.

Gogate, V. and Domingos, P. M. (2016). Probabilistic theorem proving. *Commun. ACM*, 59(7):107–115.

Golan, J. S. (1999). *Semirings and their Applications*. Springer.

Grädel, E. (1998). Description Logics and guarded fragments of first order logic. In Franconi, E., Giacomo, G. D., MacGregor, R. M., Nutt, W., and Welty, C. A., editors, *Proceedings of the 1998 International Workshop on Description Logics (DL'98), IRST, Povo - Trento, Italy, June 6-8, 1998*, volume 11 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Grädel, E., Kolaitis, P. G., and Vardi, M. Y. (1997). On the decision problem for two-variable first-order logic. *Bull. Symb. Log.*, 3(1):53–69.

Griffiths, T. L. and Yuille, A. (2012). A primer on probabilistic inference. In *The Probabilistic Mind*, chapter 2, pages 33–58. Oxford University Press.

Grove, A. J., Halpern, J. Y., and Koller, D. (1994). Random worlds and maximum entropy. *J. Artif. Intell. Res.*, 2:33–88.

Guarino, N., Oberle, D., and Staab, S. (2009). What is an ontology? In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 1–17. Springer.

Gutiérrez-Basulto, V., Jung, J. C., Lutz, C., and Schröder, L. (2011). A closer look at the probabilistic description logic Prob-EL. In Burgard, W. and Roth, D., editors, *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, pages 197–202. AAAI Press.

Gutiérrez-Basulto, V., Jung, J. C., Lutz, C., and Schröder, L. (2017). Probabilistic description logics for subjective uncertainty. *J. Artif. Intell. Res.*, 58:1–66.

Halmos, P. R. (1960). *Naive Set Theory.* Springer.

Halpern, J. Y. (1990). An analysis of first-order logics of probability. *Artif. Intell.*, 46(3):311–350.

Halpern, J. Y. (2005). *Reasoning about uncertainty.* MIT Press.

Halpern, J. Y. (2010). Lexicographic probability, conditional probability, and nonstandard probability. *Games Econ. Behav.*, 68(1):155–179.

Hastie, T., Friedman, J. H., and Tibshirani, R. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Series in Statistics. Springer.

Heinsohn, J. (1994). Probabilistic Description Logics. In de Mántaras, R. L. and Poole, D., editors, *UAI '94: Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence, Seattle, Washington, USA, July 29-31, 1994*, pages 311–318. Morgan Kaufmann.

Herbrand, J. (1930). Recherches sur la theorie de la demonstration. *Travaux de la Societe des Sciences et des Lettres de Varsovie, Class III, Sciences Mathematiques et Physiques*, 128(33).

Horrocks, I., Kutz, O., and Sattler, U. (2006). The even more irresistible SROIQ. In Press, A., editor, *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR2006).*

Horrocks, I., Patel-Schneider, P. F., and van Harmelen, F. (2003). From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26.

Hurley, P. J. and Watson, L. (2017). *A Concise Introduction to Logic*. Cengage Learning.

Jaeger, M. (1994). Probabilistic reasoning in terminological logics. In Doyle, J., Sandewall, E., and Torasso, P., editors, *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94). Bonn, Germany, May 24-27, 1994*, pages 305–316. Morgan Kaufmann.

Jaynes, E. T. (1957a). Information theory and statistical mechanics. *Physical Review*, 106:620–630.

Jaynes, E. T. (1957b). Information theory and statistical mechanics II. *Physical Review*, 108:171–190.

Jaynes, E. T. (1983). Where do we stand on maximum entropy? In Rosenkrantz, R. D., editor, *Papers on Probability, Statistics and Statistical Physics*, pages 210–314. D. Reidel Publishing Company.

Jaynes, E. T. (2003). *Probability Theory: The Logic of Science*. Cambridge University Press.

Jeffrey, R. C. (1965). *The Logic of Decision*. McGraw-Hill.

Jetti, C. R., Shaik, R., Shaik, S., and Sanagapalli, S. (2021). Disease prediction using Naïve Bayes - Machine learning algorithm. *International Journal of Science and Healthcare Research*, 6(4):17–22.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. *Comb.*, 4(4):373–396.

Katsuno, H. and Mendelzon, A. O. (1991). On the difference between updating a knowledge base and revising it. In Allen, J. F., Fikes, R., and Sandewall, E., editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91). Cambridge, MA, USA, April 22-25, 1991*, pages 387–394. Morgan Kaufmann.

Kazemi, S. M. and Poole, D. (2016). Knowledge compilation for lifted probabilistic inference: Compiling to a low-level language. In Baral, C., Delgrande, J. P., and Wolter, F., editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016*, pages 561–564. AAAI Press.

Kern-Isberner, G. (2001a). *Conditionals in Nonmonotonic Reasoning and Belief Revision - Considering Conditionals as Agents*, volume 2087 of *Lecture Notes in Computer Science*. Springer.

Kern-Isberner, G. (2001b). Revising and updating probabilistic beliefs. In *Frontiers in Belief Revision*, pages 393–408. Springer Netherlands.

Kern-Isberner, G. (2004). A thorough axiomatization of a principle of conditional preservation in belief revision. *Ann. Math. Artif. Intell.*, 40(1-2):127–164.

Kern-Isberner, G. and Brewka, G. (2017). Strong syntax splitting for iterated belief revision. In Sierra, C., editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1131–1137. ijcai.org.

Kern-Isberner, G. and Thimm, M. (2010). Novel semantical approaches to relational probabilistic conditionals. In Lin, F., Sattler, U., and Truszczynski, M., editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*. AAAI Press.

Kern-Isberner, G. and Thimm, M. (2012). A ranking semantics for first-order conditionals. In De Raedt, L., Bessiere, C., Dubois, D., Doherty, P., Frasconi, P., Heintz, F., and Lucas, P. J. F., editors, *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, pages 456–461. IOS Press.

Khemlani, S. S. and Johnson-Laird, P. N. (2012). Theories of the syllogism: A meta-analysis. *Psychological Bulletin*, 138(3):427–457.

Kimmig, A., Van den Broeck, G., and De Raedt, L. (2017). Algebraic model counting. *J. Appl. Log.*, 22:46–62.

Kisynski, J. and Poole, D. (2009). Constraint processing in lifted probabilistic inference. In Bilmes, J. A. and Ng, A. Y., editors, *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, pages 293–302. AUAI Press.

Klinov, P. (2011). *Practical Reasoning in Probabilistic Description Logic.* PhD thesis, University of Manchester, United Kingdom.

Klinov, P. and Parsia, B. (2013). Pronto: A practical probabilistic description logic reasoner. In Bobillo, F., da Costa, P. C. G., d'Amato, C., Fanizzi, N., Laskey, K. B., Laskey, K. J., Lukasiewicz, T., Nickles, M., and Pool, M., editors, *Uncertainty Reasoning for the Semantic Web II, International Workshops URSW 2008-2010 Held at ISWC and UniDL 2010 Held at FLoC, Revised Selected Papers*, volume 7123 of *Lecture Notes in Computer Science*, pages 59–79. Springer.

Koller, D., Levy, A. Y., and Pfeffer, A. (1997). P-CLASSIC: A tractable probablistic description logic. In Kuipers, B. and Webber, B. L., editors, *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, July 27-31, 1997, Providence, Rhode Island, USA*, pages 390–397. AAAI Press / The MIT Press.

Kolmogoroff, A. (1933). *Grundbegriffe der Wahrscheinlichkeitsrechnung.* Springer.

Kraus, S., Lehmann, D., and Magidor, M. (1990). Nonmonotonic reasoning, preferential models and cumulative logics. *Artif. Intell.*, 44(1):167–207.

Kuhn, H. W. (1982). Nonlinear programming: A historical view. *SIGMAP Bull., Issue 31*, pages 6–18.

Kuhn, H. W. and Tucker, A. W. (1951). Nonlinear programming. In *Proceedings of 2nd Berkeley Symposium*, pages 481–492. University of California Press.

Kullback, S. (1959). *Information theory and statistics.* John Wiley & Sons.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86.

Kunen, K. (1980). *Set Theory: An Introduction to Independence Proofs.* Elsevier.

Kuzelka, O. (2021). Weighted first-order model counting in the two-variable fragment with counting quantifiers. *J. Artif. Intell. Res.*, 70:1281–1307.

Kwong, J. M. C. (2016). Open-mindedness as a critical virtue. *Topoi*, 35:403–411.

Landes, J. (2021). A triple uniqueness of the maximum entropy approach. In Vejnarová, J. and Wilson, N., editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 16th European Conference, ECSQARU 2021, Prague, Czech Republic, September 21-24, 2021, Proceedings*, volume 12897 of *Lecture Notes in Computer Science*, pages 644–656. Springer.

Landes, J. (2023). Rules of proof for maximal entropy inference. *Int. J. Approx. Reason.*, 153:144–171.

Lang, S. (2002). *Algebra.* Springer, 3 edition.

Lee, P. M. (2012). *Bayesian Statistics: An Introduction.* Wiley, 4 edition.

Lewis, C. I. (1918). *A Survey of Symbolic Logic.* Semicentennial publications of the University of California. University of California Press.

Lewis, D. (2013). *Counterfactuals.* Wiley.

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In Nedellec, C. and Rouveirol, C., editors, *Machine Learning: ECML-98, 10th European Conference on Machine Learning, Chemnitz, Germany, April 21-23, 1998, Proceedings*, volume 1398 of *Lecture Notes in Computer Science*, pages 4–15. Springer.

Loh, S., Thimm, M., and Kern-Isberner, G. (2010). On the problem of grounding a relational probabilistic conditional knowledge base. In *Proceedings of the 14th International Workshop on Non-Monotonic Reasoning (NMR'10).*

Lukasiewicz, T. (2008). Expressive probabilistic description logics. *Artif. Intell.*, 172(6-7):852–883.

Lukasiewicz, T. and Straccia, U. (2008). Managing uncertainty and vagueness in description logics for the semantic web. *J. Web Semant.*, 6(4):291–308.

Lutz, C., Sattler, U., and Wolter, F. (2001). Description logics and the two-variable fragment. In Goble, C. A., McGuinness, D. L., Möller, R., and Patel-Schneider, P. F., editors, *Working Notes of the 2001 International Description Logics Workshop (DL-2001), Stanford, CA, USA, August 1-3, 2001*, volume 49 of *CEUR Workshop Proceedings.* CEUR-WS.org.

MacKay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms.* Cambridge University Press.

Makinson, D. (2005). *Bridges from classical to nonmonotonic logic*, volume 5 of *Texts in computing.* College Publications.

Mala, F. A. (2022). Probabilistic logic. *Indian Journal of Pure and Applied Mathematics*, 53(1):228–232.

Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Conference on Natural Language Learning*, COLING-02, pages 1—7. Association for Computational Linguistics.

Mazzoni, S. (2016). *Distribution modelling by MaxEnt: From black box to flexible toolbox.* PhD thesis, University of Oslo, Norway.

Meyer, C. and Rödder, W. (1996). Probabilistic knowledge representation and reasoning at maximum entropy by SPIRIT. In Görz, G. and Hölldobler, S., editors, *KI-96: Advances in Artificial Intelligence, 20th Annual German Conference on Artificial Intelligence, Dresden, Germany, September 17-19, 1996, Proceedings*, volume 1137 of *Lecture Notes in Computer Science*, pages 273–285. Springer.

Mintz, Y. and Aswani, A. (2017). Polynomial-time approximation for nonconvex optimization problems with an L1-constraint. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 682–687.

Montgomery, E. B. (2018). *Medical Reasoning: The Nature and Use of Medical Knowledge*. Oxford University Press.

Nathan, A., Grimberg, J., and Rhodes, A. (2023). Generative AI: Hype, or truly transformative? *Goldman Sachs: Top of Mind*, 120.

Niepert, M., Noessner, J., and Stuckenschmidt, H. (2011). Log-linear description logics. In Walsh, T., editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2153–2158. IJCAI/AAAI.

Nilsson, N. J. (1986). Probabilistic logic. *Artificial Intelligence*, 28(1):71–87.

Novák, V., Perfilieva, I., and Mockor, J. (1999). *Mathematical Principles of Fuzzy Logic*. Springer.

Nute, D. (1980). *Topcis in Conditional Logic*, volume 20 of *Philosophical Studies Series*. Springer.

Papadimitriou, C. H. (1981). On the complexity of integer programming. *J. ACM*, 28(4):765–768.

Parikh, R. (1999). Beliefs, belief revision, and splitting languages. In *Logic, Language and Computation*, volume 2, pages 266–278. Center for the Study of Language and Information.

Paris, J. B. (1998). Common sense and maximum entropy. *Synth.*, 117(1):75–93.

Paris, J. B. (2006). *The Uncertain Reasoner's Companion: A Mathematical Perspective*. Cambridge University Press, revised edition.

Paris, J. B. and Vencovská, A. (1990). A note on the inevitability of maximum entropy. *Int. J. Approx. Reason.*, 4(3):183–223.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Series in Representation and Reasoning. Morgan Kaufmann.

Pearl, J. (2009). *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2 edition.

Peñaloza, R. and Potyka, N. (2016). Probabilistic reasoning in the description logic *ALCP* with the principle of maximum entropy. In Schockaert, S. and Senellart, P., editors, *Scalable Uncertainty Management - 10th International Conference, SUM 2016, Nice, France, September 21-23, 2016, Proceedings*, volume 9858 of *Lecture Notes in Computer Science*, pages 246–259. Springer.

Pfeiffer, G. (2004). Counting transitive relations. *Journal of Integer Sequences*, 7(3).

Pratt, V. R. (1979). Models of program logics. In *20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 29-31 October 1979*, pages 115–122. IEEE Computer Society.

Presburger, M. (1929). über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Comptes Rendus du I congrès de Mathématiciens des Pays Slaves*, pages 92–101.

Ragni, M., Kern-Isberner, G., Beierle, C., and Sauerwald, K. (2020). Cognitive Logics - Features, Formalisms, and Challenges. In Giacomo, G. D., Catalá, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., and Lang, J., editors, *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2931–2932. IOS Press.

Rajendra, A. and Sajja, P. (2009). *Knowledge-Based Systems*. Jones and Bartlett Learning.

Rao, C. R. (1984). Convexity properties of entropy functions and analysis of diversity. *Lecture Notes-Monograph Series*, 5:68–77.

Reddy, K. V. R. D., Reddy, P. V., Koushik, G., and Rani, T. S. (2023). Disease prediction using Naïve Bayes. *International Journal of Research Publication and Reviews*, 4(4):2879–2885.

Reiter, R. (1980). A logic for default reasoning. *Artif. Intell.*, 13(1-2):81–132.

Richardson, M. and Domingos, P. M. (2006). Markov logic networks. *Mach. Learn.*, 62(1-2):107–136.

Riguzzi, F., Bellodi, E., Lamma, E., and Zese, R. (2013). BUNDLE: A reasoner for probabilistic ontologies. In Faber, W. and Lembo, D., editors, *Web Reasoning* *and Rule Systems - 7th International Conference, RR 2013, Mannheim, Germany, July 27-29, 2013. Proceedings*, volume 7994 of *Lecture Notes in Computer Science*, pages 183–197. Springer.

Roberts, F. and Tesman, B. (2009). *Applied Combinatorics.* Taylor & Francis, 2 edition.

Rödder, W. and Meyer, C. (1996). Coherent knowledge processing at maximum entropy by SPIRIT. In Horvitz, E. and Jensen, F. V., editors, *UAI '96: Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence, Reed College, Portland, Oregon, USA, August 1-4, 1996*, pages 470–476. Morgan Kaufmann.

Rudolph, S. (2011). Foundations of description logics. In *Reasoning Web. Semantic Technologies for the Web of Data - 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures*, volume 6848 of *Lecture Notes in Computer Science*, pages 76–136. Springer.

Rudolph, S., Krötzsch, M., and Hitzler, P. (2012). Type-elimination-based reasoning for the description logic SHIQbs using decision diagrams and disjunctive Datalog. *Log. Methods Comput. Sci.*, 8(1).

Russell, S. J. and Norvig, P. (2010). *Artificial Intelligence - A Modern Approach.* Pearson, 3 edition.

Schmidt-Schauß, M. and Smolka, G. (1991). Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26.

Schrijver, A. (1998). *Theory of Linear & Integer Programming.* Wiley.

Sebastiani, F. (1994). A probabilistic terminological logic for modelling information retrieval. In Croft, W. B. and van Rijsbergen, C. J., editors, *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)*, pages 122–130. ACM/Springer.

Shafer, G. (1976). *A Mathematical Theory of Evidence.* Princeton University Press.

Shannon, C. E. (1949). The synthesis of two-terminal switching circuits. *Bell System Technical Journal*, 28(1):59–98.

Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication.* University of Illinois Press.

Singla, P., Nath, A., and Domingos, P. M. (2010). Approximate lifted belief propagation. In *Statistical Relational Artificial Intelligence, Papers from the 2010 AAAI Workshop, Atlanta, Georgia, USA, July 12, 2010*, volume WS-10-06 of *AAAI Technical Report*. AAAI.

Sivia, D. and Skilling, J. (2006). *Data Analysis: A Bayesian Tutorial*. Oxford University Press, 2 edition.

Slater, M. (1950). Lagrange multipliers revisited. Cowles Foundation Discussion Papers 403, Yale University.

Spohn, W. (1988). A dynamic theory of epistemic states. In *Causation in Decision, Belief Change, and Statistics*, pages 105–134. Kluwer Academic Publishers.

Staab, S. and Studer, R. (2009). *Handbook on Ontologies*. Springer, 2 edition.

Stalnaker, R. C. (1976). Possible worlds. *Noûs*, 10(1):65–75.

Tarski, A., Mostowski, A., and Robinson, R. M. (2010). *Undecidable Theories: Studies in Logic and the Foundation of Mathematics*. Dover Publications.

Thimm, M. (2012). *Probabilistic Reasoning with Incomplete and Inconsistent Beliefs*, volume 336 of *DISKI*. IOS Press.

Thimm, M., Finthammer, M., Loh, S., Kern-Isberner, G., and Beierle, C. (2010). A system for relational probabilistic reasoning on maximum entropy. In Guesgen, H. W. and Murray, R. C., editors, *Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference, May 19-21, 2010, Daytona Beach, Florida, USA*. AAAI Press.

Thimm, M. and Kern-Isberner, G. (2012). On probabilistic inference in relational conditional logics. *Log. J. IGPL*, 20(5):872–908.

Thimm, M., Kern-Isberner, G., and Fisseler, J. (2011). Relational probabilistic conditional reasoning at maximum entropy. In Liu, W., editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 11th European Conference, ECSQARU 2011, Belfast, UK, June 29-July 1, 2011. Proceedings*, volume 6717 of *Lecture Notes in Computer Science*, pages 447–458. Springer.

Ullman, J. D. (1988). *Principles of Database and Knowledge-Base Systems, Volume I*, volume 14 of *Principles of computer science series*. Computer Science Press.

van Bremen, T. and Kuzelka, O. (2021). Faster lifting for two-variable logic using cell graphs. In de Campos, C. P., Maathuis, M. H., and Quaeghebeur, E., editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*, volume 161 of *Proceedings of Machine Learning Research*, pages 1393–1402. AUAI Press.

Van den Broeck, G. (2013). *Lifted Inference and Learning in Statistical Relational Models*. PhD thesis, Katholieke Universiteit Leuven, Belgium.

Van den Broeck, G. (2016). First-order model counting in a nutshell. In Kambhampati, S., editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 4086–4089. IJCAI/AAAI Press.

Van den Broeck, G., Meert, W., and Darwiche, A. (2014). Skolemization for weighted first-order model counting. In Baral, C., Giacomo, G. D., and Eiter, T., editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*. AAAI Press.

Van den Broeck, G., Taghipour, N., Meert, W., Davis, J., and De Raedt, L. (2011). Lifted probabilistic inference by first-order knowledge compilation. In Walsh, T., editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2178–2185. IJCAI/AAAI.

Van Haaren, J., Van den Broeck, G., Meert, W., and Davis, J. (2016). Lifted generative learning of markov logic networks. *Mach. Learn.*, 103(1):27–55.

Wilder, R. (1965). *Introduction to the Foundations of Mathematics*. John Wiley & Sons, 2 edition.

Wilhelm, M. (2023). Integrating linear arithmetic constraints into conditional maximum entropy reasoning. In Marquis, P., Son, T. C., and Kern-Isberner, G., editors, *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, KR 2023, Rhodes, Greece, September 2-8, 2023*, pages 700–709.

Wilhelm, M., Finthammer, M., Kern-Isberner, G., and Beierle, C. (2017a). First-order typed model counting for probabilistic conditional reasoning at maximum entropy. In Moral, S., Pivert, O., Sánchez, D., and Marín, N., editors, *Scalable Uncertainty Management - 11th International Conference, SUM 2017, Granada, Spain, October 4-6, 2017, Proceedings*, volume 10564 of *Lecture Notes in Computer Science*, pages 266–279. Springer.

Wilhelm, M., Howey, D., Kern-Isberner, G., Sauerwald, K., and Beierle, C. (2022). Integrating cognitive principles from ACT-R into probabilistic conditional reasoning by taking the example of maximum entropy reasoning. In Barták, R., Keshtkar, F., and Franklin, M., editors, *Proceedings of the Thirty-Fifth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2022, Hutchinson Island, Jensen Beach, Florida, USA, May 15-18, 2022*.

Wilhelm, M. and Kern-Isberner, G. (2017). Typed model counting and its application to probabilistic conditional reasoning at maximum entropy. In Rus, V. and Markov, Z., editors, *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2017, Marco Island, Florida, USA, May 22-24, 2017*, pages 748–753. AAAI Press.

Wilhelm, M. and Kern-Isberner, G. (2019). Maximum entropy calculations for the probabilistic description logic $\mathcal{ALC}^{\mathsf{ME}}$. In Lutz, C., Sattler, U., Tinelli, C., Turhan, A., and Wolter, F., editors, *Description Logic, Theory Combination, and All That - Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday*, volume 11560 of *Lecture Notes in Computer Science*, pages 588–609. Springer.

Wilhelm, M., Kern-Isberner, G., and Ecke, A. (2016). Propositional probabilistic reasoning at maximum entropy modulo theories. In Markov, Z. and Russell, I., editors, *Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2016, Key Largo, Florida, USA, May 16-18, 2016*, pages 690–694. AAAI Press.

Wilhelm, M., Kern-Isberner, G., and Ecke, A. (2017b). Basic independence results for maximum entropy reasoning based on relational conditionals. In Benzmüller, C., Lisetti, C. L., and Theobald, M., editors, *GCAI 2017, 3rd Global Conference on Artificial Intelligence, Miami, FL, USA, 18-22 October 2017*, volume 50 of *EPiC Series in Computing*, pages 36–50. EasyChair.

Wilhelm, M., Kern-Isberner, G., Ecke, A., and Baader, F. (2019a). Counting strategies for the probabilistic description logic $\mathcal{ALC}^{\mathsf{ME}}$ under the principle of maximum entropy. In Calimeri, F., Leone, N., and Manna, M., editors, *Logics in Artificial Intelligence - 16th European Conference, JELIA 2019, Rende, Italy, May 7-11, 2019, Proceedings*, volume 11468 of *Lecture Notes in Computer Science*, pages 434–449. Springer.

Wilhelm, M., Kern-Isberner, G., Finthammer, M., and Beierle, C. (2018). A generalized iterative scaling algorithm for maximum entropy model computations respecting probabilistic independencies. In Ferrarotti, F. and Woltran, S., editors, *Foundations of Information and Knowledge Systems - 10th International Sympo-*

sium, FoIKS 2018, Budapest, Hungary, May 14-18, 2018, Proceedings, volume 10833 of *Lecture Notes in Computer Science*, pages 379–399. Springer.

Wilhelm, M., Kern-Isberner, G., Finthammer, M., and Beierle, C. (2019b). Integrating typed model counting into first-order maximum entropy computations and the connection to Markov Logic Networks. In Barták, R. and Brawner, K. W., editors, *Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference, Sarasota, Florida, USA, May 19-22 2019*, pages 494–499. AAAI Press.

Williamson, J. (2008). Objective bayesianism with predicate languages. *Synth.*, 163(3):341–356.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3):338–353.