Ontology-Based Temporal Querying for Assessing Safety-Critical Scenarios in Road Traffic

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

Lukas Westhofen

Dortmund

2026

Tag der mündlichen Prüfung:	2.6.2026
Dekan:	Prof. Dr. Jens Teubner
Gutachter:	Prof. Dr. Daniel Neider Prof. Dr. Martin Fränzle

Abstract

Automation is moving from operating in simple, closed contexts towards open and complex situations – where one has little control over difficult-to-grasp environments involving untrained individuals, leading to increased safety concerns. Automated driving systems manifest as a notable example and serve as a motivation throughout this dissertation: The automotive industry has agreed that some residual risk during operation is unavoidable, even if rigorous verification and validation activities are performed prior. In-field observations of the system behavior are hence crucial for identifying an unreasonable residual risk, allowing the manufacturer to intervene by means of suitable safety measures. This requires in turn a thorough analysis as to what might be related to observed critical situations. However, manually inspecting time-stamped data given as video streams does not scale, especially since automated systems are anticipated to decrease economic burdens. This necessitates means to automatically query data collected in both simulated and real environments – the (aggregated) search results can then be passed to a safety engineer for further evaluation.

The goal of this dissertation is to develop a query language that allows to efficiently search in *temporal* data gathered from *complex and open contexts* to aid in safety-related data analysis. We first deepen our understanding of the steps required to mitigate risk of automated driving systems by structuring available methods. We find that a particular challenge lies in the context: In-depth domain background knowledge is required for correct data interpretation. Based on a thorough review of state-of-the-art, we argue that data assessment methods currently proposed in the literature are lacking the required semantic depth. A derivation of corresponding needs shows that Description Logics are a good fit for representing and reasoning on such knowledge, as they have already been successfully used for non-temporal ontology-based data access. The proposed query language then adds features of Linear Temporal Logic over finite traces on top of Description Logic ontologies.

The dissertation is the first to push temporal querying over expressive Description Logics towards practical applicability. Performance is of specific importance, albeit challenging given that the underlying problem is ExpTime-complete. We thus explore multiple approaches and optimizations for enabling well-performing ontology-based temporal data

access. While a first procedure compiles the temporal aspects into a finite automaton, we observe that – even in an optimized version – inefficient disjunctive queries are sometimes unnecessarily checked. A second, orthogonal algorithm therefore proposes to skip the automaton and instead operate directly on the query by means of a normal form. We are able to prove that on a non-trivial fragment, rewriting allows answering temporal queries optimally w.r.t. disjunctions, whereas the automaton-based approach is unnecessarily inefficient.

Pioneering algorithms for temporal querying over expressive Description Logic ontologies necessitates the development of novel benchmarks. We present benchmark generators for three different settings and use the resulting data for a thorough comparison of the implemented systems. Results show adequate performance of both approaches; however, using disjunctive checks more sparingly by rewriting into a normal form leads to order-of-magnitude performance improvements.

As the goal is to establish practical applicability, we finally broaden the evaluation towards embedding temporal querying into a third-party scenario-based test coverage framework. Adding the Open World Assumption inherent to Description Logics into a broader application context allows demonstrating the practical value of our language. While the underlying coverage problem becomes more complex, we can still make useful approximations under the intricacies of the real world.

"Everything is connected.
Nothing is also connected."

– Dirk Gently

Acknowledgements

While a dissertation might appear as a one-man show on the surface, it is very much not the case. Take my advisor, Daniel, who did not hesitate to supervise my project even with me having quite some groundwork already. Still, he was able to teach me much, much more in the years to come. Many other people also played a considerable role during my undertaking. Simply listing names feels inadequate. The following is therefore not only a big "thank you", but a small gift with the hope to make your day an inch better.

Most important is my family. My wife Eva accompanied me all the way. As a deeply empathetic person, she sure felt with me (nobody told me that doing a PhD would invoke such many emotions). And she enjoyed some of the perks, too: We will keep our trip to the U.S. in fond memory, where I had the fortune to present parts of this work. The birth of our son Keno quickly showed that there are way more rewarding endeavors than sitting alone in front of a screen working (or procrastinating, as many PhD students can attest to). Finally, my parents were, quite literally, my earliest supporters. The fact that this did not change in over 30 years shows that they are truly committing and caring people.

I am also grateful for the support of my friends. What comes to mind are weekends with the Kegelclub whose name is no fit for this text – making snow angels after sauna or straying on ski slopes – and spring vacations with Lennard and Basti – defying deep snow in Norway's mountains and storms in Iceland. Moreover, the winter dinners with, among others, Nick, Kerstin, Andreas, Christoph, and Johannes are annual highlights. I can still taste the Knödel(soup) of Philip Lahm's grandmother! One person shall be explicitly mentioned: Christian, as our work laid the foundation of my thesis. Initially a colleague, we became friends, sharing much in common: owning always-hungry dogs, becoming fathers, and coping with day-to-day project work (in ascending order of effort).

But it is not only the family and friends that shape the PhD student: essential for success is a facilitative work environment. Gladly, Eckard and Eike assembled such a great team. Ingo, for example, had my back several times in projects when paper writing activities appeared, mildly put, overwhelming. And of course, there's the yearly group canoe tour, where, sometimes, the impossible becomes possible!

The final acknowledgement goes to the public funds that made this work possible. I hope we never forget that such investments form the foundation of many great advances.

Contents

List of Figures

List of Tables

List of Acronyms

6LM Six-Layer Model

ACC Adaptive Cruise Control

ACI Aggregated Crash Index

ADS Automated Driving System

AFA Alternating Finite Automaton

AFGBV Autonomous Vehicles Approval and Operation Regulation

AGS Accepted Gap Size

AIS Abbreviated Injury Scale

ALKS Automated Lane Keeping System

API Application Programming Interface

ASAM Association for Standardization of Automation and Measuring Systems

ASP Answer Set Programming

A.U.T.O. Automotive Urban Traffic Ontology

BCQ Boolean combination of CQs

BTN Brake Threat Number

CI Conflict Index

CMFTBL Counting Metric First-Order Temporal Binding Logic

CNF Conjunctive Normal Form

CPI Crash Potential Index

CQ Conjunctive Query

CS Conflict Severity

CWA Closed World Assumption

DCE Distance of Closest Encounter

DL Description Logic

DNF Disjunctive Normal Form

DST Deceleration to Safety Time

FA Finite Automaton

FOL First Order Logic

GIDAS German In-Depth Accident Study

HOL Higher Order Logic

HW Headway

JVM Java Virtual Machine

KB Knowledge Base

LatJ Lateral Jerk

LongJ Longitudinal Jerk

LTL Linear Temporal Logic

LTL$_\mathbf{f}$ Linear Temporal Logic over Finite Traces

MAIS Maximum Abbreviated Injury Scale

MFOTL Metric First-Order Temporal Logic

MLTL Mission-Time Linear Temporal Logic

MTCQ Metric Temporal Conjunctive Query

MTL Metric Temporal Logic

NNF Negation Normal Form

OBDA Ontology-Based Data Access

ODD Operational Design Domain

OWL2 Web Ontology Language Version 2

PET Post Encroachment Time

PrET Predictive Encroachment Time

PRI Pedestrian Risk Index

PSD Proportion of Stopping Distance

PTTC Potential Time To Collision

ROS Robot Operating System

RSS Responsibility Sensitive Safety

SMT Satisfiability Modulo Theories

SOI Space Occupancy Index

SOTIF Safety of the Intended Functionality

STL Signal Temporal Logic

STN Steer Threat Number

StVG Road Traffic Act

StVO Road Traffic Regulation

SWRL Semantic Web Rule Language

TCI Trajectory Criticality Index

TCQ Temporal Conjunctive Query

tCQ tree-shaped Conjunctive Query

TET Time Exposed TTC

THW Time Headway

TIT Time Integrated TTC

TKB Temporal Knowledge Base

TOBM Traffic Ontology Benchmark

TSC Traffic Sequence Chart

TTB Time To Brake

TTC Time To Collision

TTCE Time To Closest Encounter

TTK Time To Kickdown

TTM Time To Maneuver

TTR Time To React

TTS Time To Steer

TTZ Time To Zebra

UCQ Union of Conjunctive Query

UN157 United Nations Regulation No. 157

UNA Unique Name Assumption

UnCQ Union of Possibly Negated Conjunctive Queries

URI Unique Resource Identifier

VTD Virtual Test Drive

W3C World Wide Web Consortium

WTTC Worst Time To Collision

1 Introduction

Contents

The pursuit for automating tasks performed by humans appears as constant throughout much of mankind's technical progress. In recent decades, it culminated in advanced cyber-physical systems aiming to reduce burdens on human operators, such as robots in industrial plants or advanced farming equipment. A commonality of both examples is their comparatively *closed context*, i.e., the automated systems operate on private property in conjunction with trained personnel. In recent years, we observed a push towards more *open contexts*, where systems are placed in uncontrolled environments in which, e.g., lay persons can directly reach the vicinity of the machine. While progress has been made in automating human tasks in open yet (comparatively) simple contexts, such as subway autopilots, the issue of handling an open *and* complex context appears as a Herculean task. Examples for the latter setting include automated systems for controlling maritime vessels in ports, urban drone deliveries, home-assistant robots, and road vehicle driving. This dissertation is particularly concerned with Automated Driving Systems (ADSs) for road traffic, where naturally the context is complex – as anyone who has undergone a practical driving exam can attest – as well as open – interactions with animals, unskilled humans, and unexpected objects can occur at any time.

It is partially due to the open and complex context that ensuring safety of such systems has proven to be a major roadblock to their public release; even a special term was

coined for this: “approval trap” [WW16]. In a nutshell, it is not economically feasible to demonstrate statistically acceptable safety prior to type approval by randomly testing the machine in the field. Current industrial practices therefore employ a two-pillar approach: maximizing the certainty on the estimation of risk introduced by the system before its release based on a set of assumptions, and continuously validating the system's behavior and the assumptions in the field. Take, for example, Waymo, whose vehicle fleet has driven a cumulative 56.7 million miles in public testing campaigns as of January 2025 [Kus+25] as a contribution to the second pillar. In order to fully make use of data collected both prior to and post deployment, we require in-depth data analysis methods: How many critical events were there? Were they induced by hazardous behavior of the system? In which type of situations does this behavior occur? These analyses are recommended by ISO 21448 [Int22], which in turn forms the basis for an assessment compliant with national and international homologation regulations, such as UNECE R157 [Uni21].

It must be concluded that a powerful means to analyze the produced data are required. But how could such an instrument look like? Let us examine the task at hand. Each sampling iteration, the system produces raw sensor data, which are processed to a more abstract representation, e.g., three-dimensional bounding boxes of surrounding objects. By repetition, milliseconds after milliseconds, a temporal sequence of states is assembled. In case of a critical event (such as a near-miss encounter) or simply as a precaution, a history of the recently recorded sequence is sent away for further investigation – to extract factors from the temporal sequence that, if seen often enough, can point engineers to implementation faults or unsafe specifications.

However, manual analysis efforts are bound to fail due to the sheer size of the records: Waymo, for example, reportedly operated over 450 000 passenger trips a week as of December 2025 [WB25]. Such data must be analyzed in an automated fashion. A straightforward solution is to store data in an off-the-shelf time series database and apply queries to find, for example, suspicious accumulations of certain factors in critical situations. However, we are confronted with a complex context, meaning that implicit background knowledge might be required for validly answering queries: If, for example, the user asks for all critical situations in which a human was involved, then all instances where the system's classifier labeled objects as children as well as adult pedestrians shall be returned. Standard query languages for time series databases do not offer such features but instead interpret the query directly over the data.

An established solution to incorporating background knowledge (such as “children and adult pedestrians are humans”) is executing the query not only over the data but also an ontology – a formalized domain conceptualization. While many ways of formulating ontologies have been proposed, so-called Description Logics (DLs) emerged in the 1980s and established themselves as the predominant formalism since then [Baa+07]. Already early on, in the 1990s, their potential for enhancing established techniques from the

database community was understood [Cal+98]. Specifically, Conjunctive Queries (CQs), a first order language with conjunction and existential quantification, were found to strike a balance between expressiveness and computational complexity [HT00]. This led to the paradigm of Ontology-Based Data Access (OBDA), where data are retrieved under an ontology, which was successfully used for querying non-temporal data over ontologies, e.g., at Statoil [Kha+17].

When confronted with temporal data, it appears obvious that OBDA can be extended by temporal features. The 2000s and 2010s gave birth to two major lines of research on answering temporal queries over DL ontologies: First, since computational complexity of query answering was deemed unsuitable for many practical applications, often, DL fragments with restricted expressiveness were studied [AMO18]. Many efficient tools, such as Ontop-temporal [Kal+18] and MeTeoR [Wan+22], have been implemented. The second major line is concerned with highly expressive DL fragments and analyzed various theoretical aspects of temporal querying [Lip14]. However, up to now, the setting of temporal querying over expressive DLs has neither been examined from a practical perspective nor been implemented.

This dissertation argues that the expressive setting is a valuable addition for applications requiring extensive background knowledge in reasoning. In essence, the domain knowledge required for correctly analyzing data is already complex in itself; consider, e.g., the legal and regulatory details on the constituents of roads and intersections [Str06] and their implications on traffic rules [Wes+22b]. Such domain knowledge must be preserved in the data analysis process, otherwise results might be faulty and wrong conclusions are drawn. This work thus looks only at expressive DLs and does not further examine temporal querying under restricted DL expressiveness. For example, if we use a simple road network model neglecting essential proportions of the rules as to what counts as a lowered curb, an automated analysis of the right of way situations is deemed to yield wrong answers. This can have severe downstream effects: The engineers might not be able to correctly link an accumulation of critical scenarios to the underlying fault (for example, an incorrect right of way implementation). This motivates the need for requiring expressive background knowledge when temporally querying data for open and complex contexts; simultaneously, this topic has – to the day of writing – only be examined from a theoretical perspective. Therefore, the following central statement is the subject of this dissertation:

> "Temporal querying in expressive DLs can be made practically applicable, specifically for assessing safety-critical scenarios in road traffic."

1.1 A Reader's Guide to the Dissertation

In order to argue in favor of the above statement, we first require an understanding of "practically applicable." As the dissertation title and the prior motivation suggests, this work is embedded in a practical context for exactly that reason. In Chapter 2, we thus examine how temporal data are currently analyzed for assessing ADSs. Here, ADSs act as a representative instance of automated systems operating in open and complex contexts. The chapter derives gaps on which data analysis mechanisms can aid practitioners in raising the quality of their results but are currently unavailable in their toolbox.

These gaps are the basis for devising a solution in Chapter 3 – a formal language called MTCQs – that aims to advance the support of practitioners in temporal data analysis. We first dive into the foundations upon which MTCQs are built. An examination of existing literature regarding related querying mechanisms enables us to delineate MTCQs from the state-of-the-art and highlights why current approaches are insufficient as a theoretical foundation for bridging the aforementioned gaps. This leads to the presentation of the MTCQ formalism itself, including relevant problems.

The core of this dissertation lies in Chapter 4, which contributes novel methods to tractably answering MTCQs over expressive DLs. We expand some of these ideas, leading to a first, optimized algorithm based on Finite Automata (FAs). Since practical observations show that it still does not behave ideal in some cases, we propose a second algorithm. Here, instead of FAs, rewriting rules are employed as to avoid wasteful behavior.

It remains for Chapter 5 to deliver evidence for the claim of practical applicability. Both algorithms are implemented in a state-of-the-art DL reasoner, enabling an evaluation based on the application domain of automated driving. The goal is to demonstrate two essential properties of our approach: First, performance is reasonably sufficient under realistic temporal data and background knowledge. Second, it provides value to the practitioner.

The thesis is thus structured along four fundamental steps: extracting a meaning for "practically applicable" (Chapter 2), devising a suitable temporal query language from that (Chapter 3), showing how such queries can be tractably answered (Chapter 4), and substantiating the claim of practicability (Chapter 5). What follows is a short guide through each chapter with the intention to facilitate their navigation and deliver all essential information to the reader.

Assessing Safety-Critical Scenarios in Road Traffic The dissertation at hand looks at temporal querying through the lens of safeguarding ADSs – as one example of safety-critical systems taking over demanding tasks while operating in open and complex contexts. The purpose of this exercise is to anchor all subsequent delineations in the demands of practitioners. Therefore, Chapter 2 assesses how temporal querying embeds into

contemporary research on ADS safety. The goal of this chapter is to understand and, if necessary, improve current industry practices by critically reviewing the state-of-the-art.

We start by assuming a broad view on ADS safety in Section 2.1, specifically, on what has emerged as Safety of the Intended Functionality (SOTIF) in ISO 21448 [Int22]. The central aspect of this standard is the absence of unreasonable risk originating from hazards caused by the specified functionality. Section 2.1 explains the terminological framework of ISO 21448 and, since several inconsistencies are present in the standard, constructively provides improvements [Put+23]. The overall goal is to show that the system is safe. For this, quantification is suggested in Annex C of ISO 21448, which is subsequently delineated and for which again several improvements are made. Overall, this consistent terminology, combined with a well-defined quantification approach, lays the foundation for a feasible SOTIF assessment.

Two missing ingredients for conducting SOTIF risk quantification in practice are discussed in Section 2.2. First, data must be *generated*, which is the topic of Section 2.2.1. For this, scenario based testing has emerged as a promising technique. It recognizes that, as systems operate over time, input streams of finite but essentially arbitrary lengths make it difficult to effectively argue for safety. Instead, it proposes to break the inputs down into smaller, more manageable chunks of temporal sequences – called scenarios [Neu+20] (hence the name). They essentially present a sequence of actions in which a specific task is to be performed by the system, such as crossing an intersection, overtaking other cars, or simply driving straight for a while. After the task is concluded, the scenario ends (and another might start). One can imagine scenarios as the "currency" used in all safeguarding activities. Executing the system in a given scenario is referred to as scenario-based testing. Many techniques for this exist, including various simulation-based sampling methods, which are briefly presented to gain an understanding of how the generated data are shaped [Neu+20].

The second required ingredient for SOTIF evaluation is examined in Section 2.2.2, namely, how these data are *analyzed*. We differentiate between two kind of factors that are desirable to assess: those approximating harm or severity, e.g., by measuring close distances, and factors causally preceding harm, e.g., low visibility in a crowded traffic situation.

The most prominent representatives of the first class are so-called criticality metrics, which aim to quantify various aspects of risk within recorded scenarios. A plethora of metrics were proposed since the 1970s, including the well-established Time To Collision (TTC), Required Acceleration (a_{req}), and Post Encroachment Time (PET) [Wes+23, Section 5.2]. These act as proxies for harm or severity and hence possess the ability to lower the burden on the amount of collected data (since critical events appear more often than actual accidents). Yet, this shifts efforts towards assessing validity of these metrics; and hence a critical review of the state-of-the-art is presented.

The second class of factor analyzed in data for SOTIF evaluation concerns events in road traffic preceding critical situations. These include hazardous behavior of the system (such as violating the safety distance), but also conditions triggering this behavior (such as low visibility) as well as any other factor increasing criticality (such as being behind a vulnerable road user). While criticality metrics are heavily focused on physical quantities, these factors are concerned with abstract relations between entities in traffic. By means of formalization, such factors can be made determinable in data, with the advantage of having an unambiguous semantics. Hence, various logic-based proposals were made in recent years, many based on temporal logics [Hül+10; Luc+16; Els+20; Gel+20]. We give an overview of this literature to conclude our review on the state-of-the-art.

Based on both state-of-the-art reviews – for criticality metrics and logics-based data querying – we draw the following conclusion: Criticality metrics are already well-researched due to its roots in traffic conflict analysis in the 1970s, and a plethora of proposed metrics are readily available to the practitioner. However, logics-based data querying for purposes of safety analysis is still under research and does not satisfy all practical needs. Specifically, all logic-based approaches have in common that they employ no or only comparatively inexpressive formalisms to model the system's environment (such as propositional logic). They therefore do not adequately account for the complexity of the environment. In Section 2.3, we hence derive what is missing from the currently existing research. These gaps in the current formalisms for specifying indirect criticality proxies allow us to identify requirements on what a valuable addition for the practitioner's toolbox might look like – and, as we will later see, form the foundation for our custom temporal query language. Essentially, the chapter concludes that we require logics-based searching in temporal data enhanced by a formal representation of the rich domain knowledge available to engineers.

A Temporal Query Language An established solution for formalizing domain knowledge are DLs, which, due to their popularity, will be the only ontology formalism examined in this dissertation. Here, inclusion relations between axioms over a set of concepts and roles describe valid states of the world. The simplest form of a DL ontology is a *taxonomy*, where only inclusions like $\texttt{Pedestrian} \sqsubseteq \texttt{Human}$ (any pedestrian is a human) are allowed. However, DLs also allow for more complex composite axioms like $\texttt{Driver} \sqsubseteq \texttt{Human} \sqcap \exists \texttt{drives}.\texttt{Vehicle}$ (any driver is a human and must drive some vehicle). Here, `Driver`, `Human`, and `Vehicle` are concepts and `drives` is a role (a binary relation between concepts). A set of such inclusion axioms forms a DL ontology, which we denote by $\mathcal{O}$.

Rich feature sets have been proposed over the last decades, and the resulting DL fragments were named accordingly. For example, $\mathcal{ALC}$ denotes the Attributive Language with Complements, which allows for union, intersection, negation, and existential and universal quantification along roles. The sophisticated DL $\mathcal{SROIQ}^{(\mathcal{D})}$, which is a superset of $\mathcal{ALC}$, forms the basis of the World Wide Web Consortium (W3C) standard Web Ontology Language Version 2 (OWL2) [Wor12], and is presented as the formal foundation

of this work in Section 3.2.

As motivated earlier, a DL ontology can be used to reason over a set of data $\mathcal{D}$. In DLs, the data are modeled as assignments of concepts and roles to so-called individuals, which are observable entities in the world. For example, we model a perceived driver `d` as `Driver(d)` in the data. The combination of both ontology and data is a Knowledge Base (KB) $\mathcal{K} = (\mathcal{O}, \mathcal{D})$. While various reasoning services over KBs have been proposed and examined (such as consistency, instance retrieval, or subsumption), this dissertation is concerned with *query answering*. The idea is to allow the user to formulate queries, comparable to standard database retrieval, which have to be answered. The answering procedure returns lists of individuals from the data that satisfy the query's constraints. A standard query language is CQs, which enables the user to ask for individuals that satisfy a set of membership constraints to concepts or roles, with an additional feature of existential quantification. It has been found that CQ answering can be heavily optimized in practice. In Section 3.2, we take a look at some of the state-of-the-art techniques explaining why CQ answering is tractable, even when the underlying decision problem is ExpTime-hard.

To extend querying from only a single set of data to time-stamped sequences of data, temporal query languages were subsequently developed. They can, for example, resemble Linear Temporal Logic (LTL) and use CQs in place of propositions [Lip14]. Typically, we assume a discrete, finite history of already gathered data for some time points 0 to n, which, together with the ontology, form a Temporal Knowledge Base (TKB) $\mathcal{K} = (\mathcal{O}, (\mathcal{D}_i)_{i \in \{0,\dots,n\}})$. The ontology additionally restricts an infinite sequence of possible future states – hence, the resulting logic is similar to LTL with past operators. Besides Temporal Conjunctive Queries (TCQs), an array of different temporal querying formalisms (also over different DLs fragments) were proposed [GK12; ÖM14]. A detailed listing is provided in Section 3.3.

For the final part of the chapter, Section 3.4, we leverage this prior work to derive a custom temporal query language (called MTCQs) specifically tailored towards the previously sketched application context. It is distinct in two features: First, from the practical perspective, it can be argued that analyzing the historical samples is already sufficient, and we therefore interpret MTCQs solely over discrete, finite data. The application context hence enables us to disregard, for the scope of this dissertation, continuous or infinite data. The proposed language hence allows for the usual temporal operators for finite sequences, such as until ($\mathcal{U}$), eventually ($\Diamond$), globally ($\Box$), as well as weak and strong next ($\bullet$, $\bigcirc$). Second, it must be assumed that interval durations play important roles , e.g., to decide whether an overtaking maneuver takes too long, which is why MTCQs additionally allow metric operators, such as a time-bounded globally ($\Box_{[a,b]}$). With MTCQs, we can, for example, ask for every individual x that drives some vehicle

for at least two time points by

$$\Phi(x) = \Diamond\Box_{[0,1]}\exists v.\texttt{drives}(x, v).$$

We are mainly interested in query answering over MTCQs. Here, for a given TKB, we ask for assignments from the individuals to the free variables of the query for which, when acknowledging the ontology, we can find a match for the query in the data. Take, as an example, a TKB over the already introduced driver axiom and these three observations: first, we observe nothing and in the two subsequent time points, there is some driver. Formally, this is expressed as

$$\mathcal{K} = \big(\{\texttt{Driver} \sqsubseteq \texttt{Human} \sqcap \exists\texttt{drives.Vehicle}\}, (\emptyset, \{\texttt{Driver(d)}\}, \{\texttt{Driver(d)}\})\big).$$

The single answer to the MTCQ $\Phi(x)$ assigns the individual $\texttt{d}$ to the only free (answer) variable x – since it matches the necessary existence of a driven vehicle through the background knowledge of the ontology for the last two time points.

Tractable Temporal Querying over Expressive Description Logics Despite (or due to) the practical relevancy of MTCQs, answering them is not easy. Even for MTCQs without any variable, i.e., checking whether the given data matches the query under the ontology, the problem is ExpTime-complete for $\mathcal{ALC}$. The lower bound follows from the foundational problem of checking consistency of a TKB, and luckily, the upper bound coincides with this lower bound. However, ExpTime-complete problems are notoriously intractable in practical settings, especially under real-world inputs. This holds specifically in our case, algorithms can have an exponential dependency not only on the query and ontology, but also on the data, which we must assume to be large. The main question of Chapter 4 is therefore: How can we efficiently answer MTCQs? We present two different approaches.

Up to now, no practical algorithm has been proposed for answering temporal queries over expressive DLs. As mentioned earlier, the only work in this area examines a related setting from a theoretical view [Lip14]. The upper bound proof yields a highly non-optimized, exponential algorithm, which is based on Büchi automata. For the first approach, we adopt this strategy and propose an FA-based algorithm [Wes+24a]. In its core, the technique represents the temporal constraints as an automaton, where state transitions are encoded by satisfiability checks of disjunctions of CQs, so-called Unions of Conjunctive Queries (UCQs). For this algorithm, Section 4.1 provides proofs of correctness and run time. Since a naïve implementation would not satisfy the requirements of the application context, Section 4.1 proposes a foundational optimization that extracts information from CQ answering to aid in the required satisfiability checks.

It turns out that by using a standard translation of Linear Temporal Logic over Finite Traces (LTL_f) into FAs, sometimes unnecessary UCQs are checked (even with the

foundational CQ optimization). In these cases, checking the disjunction is, in fact, not required, and it would suffice to simply answer CQs. Alas, answering UCQs in expressive DLs is, in practice, more expensive and less optimizable than it is for CQs – it is therefore advantageous to rely on CQ answering as much possible. In Section 4.1, we conclude that, while the proposed algorithm shows theoretically optimal behavior, it does not behave optimal in a practical sense.

For this reason, a second algorithm is proposed in Section 4.2. It relies on the observation that unnecessary UCQs arise from the standard translation from $\mathrm{LTL_f}$ to deterministic FAs. We thus ditch the use of automata completely. The novel algorithm assumes a rewriting framework instead. Here, the query is rewritten into a normal form that allows inductively answering subqueries until a base case – a nontemporal query – is reached. We again provide correctness and complexity proofs.

We continue with a theoretical examination of the difference in UCQ handling between both algorithms. On an example, we can easily establish that the second algorithm exhibits much more benign behavior w.r.t. CQ and UCQ answering. The question arises whether we can generalize this issue. In Section 4.3, we are able to extract a non-trivial fragment of MTCQs for which no UCQ checks are, in general, needed. This is not always trivial to see: Consider, for example, the MTCQ

$$\Phi(x) = \Diamond(\texttt{Pedestrian}(x) \wedge \bigcirc\texttt{Human}(x)).$$

Here, an implicit disjunction arises due to the combination of $\Diamond$ and $\bigcirc$ – for the second time point, it has to be checked whether $\texttt{Human}(x)$ or $\texttt{Pedestrian}(x)$ holds. Therefore, UCQs are actually required for correctly answering $\Phi(x)$. Since this property is not directly evident from a manual inspection, the final part of Chapter 4 gives a formal and efficiently decidable characterization for queries in which no UCQ answering is required. We will see that the second approaches operates, in fact, optimal w.r.t. UCQs on the specified fragment. The first approach, on the other hand, induces UCQ checks on many of the queries in this fragment. We conclude the chapter with a theoretical analysis on the reasons behind this behavior. In essence, it is due to using deterministic FAs, as we are able to show that, for each query in the fragment, there is always a nondeterministic FA that allows answering the query without UCQs.

Demonstrating Practical Applicability The attentive reader will have noticed some assertive claims in the previous paragraphs: The second algorithm is more frugal w.r.t. UCQ answering; this to have practically observable effects on performance; and, in general, that both algorithms can be used in practical settings. The evaluation presented in Chapter 5 is hence in order. First, the software tool Topllet – implementing both algorithms – is presented in Section 5.1 [Wes+24b]. It is integrated in the DL reasoner Openllet and uses a custom extension of OWL2 to represent TKBs.

The goal of the tool is to provide evidence for the aforementioned claims in Section 5.3. Since Topllet is, to the best of our knowledge, the first implementation of any temporal reasoning tool over expressive DL ontologies, no benchmark exists. Therefore, the first task is to construct benchmarks serving two different purposes: One for lightweight, efficient evaluation during development, and a large-scale benchmark that mimics valid real-world data while allowing for scalability in the data size. We again retreat to our example application of data analysis for automated driving. For the first benchmark, an example from the German traffic act – giving right of way to two-wheelers – is formalized in a lean $\mathcal{ALCRQ}$ ontology. The second benchmark uses the Automotive Urban Traffic Ontology (A.U.T.O.) as an established $\mathcal{SRIQ}^{(\mathcal{D})}$ ontology [Wes+22a], since the goal is to be as close to real-world applications as possible. A.U.T.O. is based upon the so-called 6-Layer Model, an industrially adopted scheme for categorizing urban road traffic [Sch+21], and contains a wide array of concepts and relations between entities relevant in traffic. The resulting benchmark set – called the Traffic Ontology Benchmark (TOBM) [Wes+24a] – simulates traffic-participant behavior on two different intersections in a two-dimensional setting.

Running Topllet on TOBM yields two major insights: First, both algorithms can be applied in practically relevant offline analysis settings. Second, the rewriting based approach performs approximately one order of magnitude better than the FAs-based technique. The evaluation shows that it is mostly due to the more effective usage of CQ answering, as claimed earlier. On top of this, for some queries, performance improvements enable running Topllet in online settings, i.e., answering the query over a data stream.

While Section 5.3 evaluates performance, it remains to show that MTCQ answering provides value in an applied sense, meaning that engineers can benefit from tools like Topllet. It is not within the scope of this dissertation to conduct a large-scale, empirical usability study. We will, however, demonstrate in Section 5.4 that Topllet can be successfully integrated into a scenario-based test coverage framework, an example application context in which temporal logics are widely used. Classical temporal logics, however, assume a closed world, which is not always justifiable from a practical perspective. We show that using the open world of MTCQs avoids this problematic assumption, but comes with far-reaching effects: Rigorous adaptions of test coverage theory and implementation are necessary. But, fortunately, even under the semantics of MTCQs, statements on coverage can still be made, as we show in an evaluation on synthetic and real-world data. This successful integration of Topllet into a broader SOTIF-context exemplifies that the language and algorithms presented in this dissertation add practical value.

1.2 Contribution

By now, the contribution of this dissertation should be apparent. To recapitulate, it embeds temporal querying over expressive DLs in a practical context, gives novel algorithms for tractably answering such queries, and demonstrates their practical applicability in the embedded context. The following works were published in support of this contribution:

1. Using Ontologies for the Formalization and Recognition of Criticality for Automated Driving, Lukas Westhofen, Christian Neurohr, Martin Butz, Maike Scholtes, Michael Schuldes, 2022, *IEEE Open Journal of Intelligent Transportation Systems* [Wes+22a].

2. Criticality Metrics for Automated Driving: A Review and Suitability Analysis of the State of the Art, Lukas Westhofen, Christian Neurohr, Tjark Koopmann, Martin Butz, Barbara Schütt, Fabian Utesch, Birte Neurohr, Christian Gutenkunst, Eckard Böde, 2023, *Archives of Computational Methods in Engineering* [Wes+23].

3. On Quantification for SOTIF Validation of Automated Driving Systems, Lina Putze, Lukas Westhofen, Tjark Koopmann, Eckard Böde, Christian Neurohr, 2023, *IEEE Intelligent Vehicles Symposium (IV)* [Put+23].

4. Answering Temporal Conjunctive Queries over Description Logic Ontologies for Situation Recognition in Complex Operational Domains, Lukas Westhofen, Christian Neurohr, Jean Christoph Jung, Daniel Neider, 2024, *International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)* [Wes+24a].

5. Topllet: An Optimized Engine for Answering Metric Temporal Conjunctive Queries, Lukas Westhofen, Christian Neurohr, Jean Christoph Jung, Daniel Neider, 2024, *NASA Formal Methods Symposium (NFM)* [Wes+24b].

6. Temporal Conjunctive Query Answering via Rewriting, Lukas Westhofen, Jean Christoph Jung, Daniel Neider, 2025, *Annual AAAI Conference on Artificial Intelligence (AAAI)* [WJN25].

7. Test Coverage of Automated Robotic Systems in Open World Environments, Lukas Westhofen, Till Schallau, Dominik Schmid, Stefan Naujokat, Falk Howar, Daniel Neider, 2026, *International Symposium on Formal Methods (FM)* [Wes+26].

Two published articles contain parts that additionally contributed to the dissertation at hand:

1. Fundamental Considerations around Scenario-Based Testing for Automated Driving, Christian Neurohr, Lukas Westhofen, Tabea Henning, Thies de Graaff,

Eike Möhlmann, Eckard Böde, 2020, *IEEE Intelligent Vehicles Symposium (IV)* [Neu+20].

2. 6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment, Maike Scholtes, Lukas Westhofen, Lara Ruth Turner, Katrin Lotto, Michael Schuldes, Hendrik Weber, Nicolas Wagener, Christian Neurohr, Martin Herbert Bollmann, Franziska Körtke, Johannes Hiller, Michael Hoss, Julian Bock, Lutz Eckstein, 2021, *IEEE Access* [Sch+21].

This dissertation incorporates selected content of all mentioned publications. At the beginning of each chapter, appropriate references are given. The contribution of the respective publication to the chapter is highlighted and put into the context of this dissertation.

Doctoral regulations require explicating my own contribution to the above works. These papers are a collaborative work of over three dozen authors from a variety of institutions. I am very grateful for their provided efforts, inputs, improvements, and guidance. As it is often the case, ideas are born in discussions, and exact attribution is not always possible in hindsight. Moreover, the listing of an activity in this document does not exclude the possibility that it was also performed by co-authors. Taking these notes into account, the following paragraphs present an attempt to qualify my contributions to the mentioned publications.

Fundamental Considerations around Scenario-Based Testing for Automated Driving Shortly after starting my work at what was back then the transportation department at OFFIS, my coworker Christian assembled a team to come to terms with common yet implicit assumptions in scenario-based testing. Work on this paper was naturally a team effort, and I was involved in many of the discussions on extracting and writing up several fundamental challenges. My specific focus at that time concerned scenario and requirement elicitation. While already having experience with scientific publishing in formal methods thanks to my excellent bachelor and master theses supervisors Nils and Jana, this work offered a great entry point to the domain of automated driving safety: It allowed me to perform extensive literature reviews, some of which found their way into the paper. The lively discussions and the refreshing approach to questioning widely accepted assumptions strongly shaped my way of working.

6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment My early work on this dissertation was heavily influenced by the publicly funded research project VVMethods. One particularly thought-provoking topic was the so-called "layer model," which prior projects and publications used in various flavors. As our ontological work in the project was ramping up, we decided that the layer model should act as a semantic foundation. A unification of all previous proposals was required;

otherwise, no common ontology could be achieved. I provided an initial draft to the working group, but swiftly, the political dimension of the undertaking became apparent. In order to integrate the views of all involved parties, Maike and Lara thankfully took over the main lead on this activity. I continued to provide feedback on several drafts, and accompanied Maike and Lara to the finish line.

Using Ontologies for the Formalization and Recognition of Criticality for Automated Driving Having led the ontology working group in VVMethods, it was only reasonable to polish our work for publication. The initial goal was to simply publish the ontology, for which I acted as the lead developer. It became clear that a demonstration of its usefulness was needed. Two components came together almost seamlessly: I was curious to test out the applicability of available reasoning tools on the ontology. Our collaborators at the Institute for Automotive Engineering at the RWTH Aachen were currently assembling data in a rich domain model. Both combined resulted in what is presented in the publication: I developed a software tool chain that is able to reason on our ontology and the provided data to identify critical scenarios. The subsequent write-up and experiments were led by me, and the paper kick-started this dissertation.

Criticality Metrics for Automated Driving: A Review and Suitability Analysis of the State of the Art This publication is probably the most collaborative work I have participated in yet. Christian and I originally collected several criticality metrics for an internal presentation, and quickly noticed that no unified overview was available in the literature. Dissatisfied with the situation, we decided to assemble a diverse team of nine authors from the VVMethods and SET Level projects to address the issue. While I acted as the organizational lead, structured the work, and guided the writing activities, each contributor brought their own criticality metrics to the table. Putting these into a combined framework was challenging and only possible with teamwork. The resulting website has received positive feedback, and was set up by me based on an impulse by Tjark, a coworker of mine.

On Quantification for SOTIF Validation of Automated Driving Systems After reading a draft of the (back then) ISO/PAS 21448, it quickly became apparent that the upcoming standard highly impacts this dissertation. My colleagues Christian, Tjark, and Lina originally set out to conduct a hazard analysis and risk assessment according to the then published ISO 21448. I provided support by specifying, together with Lina, a system model to consider for further analysis. It quickly turned out that interpreting the standard literally was impossible – were we the first to actually implement a standard-compliant hazard analysis? We abandoned our hopes of executing the original plan and focused on fixing the identified inconsistencies instead. This meant several exhausting group sessions of reading the document and assembling a new, consistent mental model. I mainly

focused on the terminological side, where I supported Lina in untangling the intertwined definitions and coming up with improved examples and figures. During writing the paper, I reviewed and refined several drafts for many of the sections.

Answering Temporal Conjunctive Queries over Description Logic Ontologies for Situation Recognition in Complex Operational Domains My earlier work on using ontologies on automotive data was driven by available tooling. Daniel (my advisor) and I revisited it from a more foundational perspective. Based on a literature review conducted by me, a promising framework emerged, for which, however, neither a practical algorithm nor an implementation was available. I therefore immersed myself, for quite some time, into the development of a custom temporal querying tool, guided by the theoretical experience of Daniel and the practical view offered by Christian. Additional expertise on the theory of DLs came in the form of Jean, thanks to whom we were able to clearly work out several theoretical results. After multiple development iterations, the novel benchmarks I created finally showed satisfying results. Writing was again a team effort, for which I took the main responsibility under the extensive guidance of my co-authors.

Topllet: An Optimized Engine for Answering Metric Temporal Conjunctive Queries As the above work described the temporal querying algorithm from a theoretical view, the team of authors decided that a separate tool paper was in order, with the goal of highlighting its inputs and usage as well as presenting insights into the practical implementation. Due to me being the only developer of the software, I took over creating and evaluating new benchmarks, coming up with easily understandable examples, and large parts of writing the paper. All these activities were constantly guided and refined by my collaborators.

Temporal Conjunctive Query Answering via Rewriting I spent quite some time trying to iteratively improve the performance of the implemented querying algorithm, with mixed results. A fresh approach was desperately needed. After trying out several possibilities, I came up with a promising rewriting procedure. I quickly set out to implement a rough version of the envisioned algorithm; first benchmarks showed drastic performance improvements. Inspired by the promising results, I hastily assembled some slides, took a train to Dortmund, and presented them to both Daniel and Jean, who gave valuable feedback and several lines of improvement. I continued development until a publishable state was reached, and the team of authors set out to write down the results. The writing activities were again lead by me, including proofs for the theoretical results, but many important revisions were made by Daniel and Jean.

Advancing Test Coverage of Automated Robotic Systems to Open World Environments During times when processors produced quite some heat just to run my software,

I often questioned the use of it all: Is this high computational complexity even worth it? What can be gained by using my tooling from a practical perspective, compared to available temporal logic software? Gladly, an opportunity of investigating these questions presented itself when I met Till, also a PhD student at the TU Dortmund University, in California. We decided that a collaboration would be of great interest to both of us. The weeks after, I started investigating the possible implications of integrating my work into Till's broader test coverage framework, and came up with the idea of having "unknowns" in the scenario data. I presented the concept to Till and his colleagues, who showed keen interest. A first prototype was implemented by me, but exhibited poor performance. This did not deter us from exploring other options, and I started experimenting with MaxSAT- and graph-based algorithms. These implementations finally showed success, and Dominik, a coworker of Till, set out to implement these algorithms into their software, further improving performance. Satisfied with the results, I took over the main responsibility for writing and coordinating our efforts. Specifically, I formalized the novel coverage framework, devised the two corresponding problems, and gave proofs for the theoretical results. When time was ripe to evaluate our approach, Dominik used synthetic data to put the algorithms through their paces, while I provided an analysis of inD to validate our assumptions on the probabilities of "unknowns" in real-world data.

2 Assessing Safety-Critical Scenarios in Road Traffic

Contents

We start our journey of pushing temporal querying over expressive ontologies towards practical usability by analyzing an industrially relevant application context: safeguarding automated driving. After a thorough revision of risk quantification proposed in ISO 21448, one of the two central standards in ADS safety, (Section 2.1) we review contemporary literature on methods that can support in this quantification (Section 2.2). From this, we identify what is currently missing from the perspective of temporal querying in Section 2.3. This not only grounds the language and algorithm development in Chapter 3 and Chapter 4 as well as the evaluation in Chapter 5 in practice, but also contributes to the scientific state-of-the-art. The chapter is based on two published literature reviews [Neu+20; Wes+23] and a study revisiting ISO 21448 [Put+23], all of which were one of the first publications in their respective fields.

The first contribution of this chapter, given in Section 2.1, is a critical review and revision of ISO 21448 [Int22], an essential standard in safeguarding ADSs. In contrast to

established automation features, such as Adaptive Cruise Control (ACC), ADSs cannot assume an alert human fallback operator available at any time. This missing supervision makes having a safe specification imperative – after all, the system itself must handle critical situations. This has been recognized with the introduction of SOTIF in the ISO 21448 standard, which has been published in 2022. It has since then become one of two major normative pillars in ADS safety – the other one being ISO 26262 [Int18], which complementarily assumes a safe specification and examines technical faults leading to specification violations. While the preceding decade has shown that adhering to ISO 26262 on an industrial scale is technically feasible [Pal+11; Dar+12], demonstrating safety – i.e., the absence of unreasonable risk – according to ISO 21448 remains challenging [Ber+22; Wan+24b]. We hence make SOTIF the central motivation of this dissertation. Its introduction is based on the work of Putze, Westhofen, Koopmann et al. [Put+23], who, in 2023, spotted significant inconsistencies in the risk framework proposed in the standard. Hence, this dissertation presents a revised, consistent perspective on SOTIF and its quantification approach.

While ISO 21448 demands to demonstrate the achievement of SOTIF, e.g., by quantification, it gives only abstract guidance on how to estimate the required quantities, which is the topic of Section 2.2. In the past, the automotive industry has relied on distance-based methods, where manufacturers drive a manageable distance on proving grounds and in real world test drives. For ADSs, these methods are infeasible due to the enormous driving distance that has to be covered [KP16; WW16]. Section 2.2.1 presents an envisaged solution: Scenario-based testing, where the the system is executed in varying environment conditions for some finite time. The review is based in parts on the work by Neurohr, Westhofen, Henning et al. in 2020 [Neu+20], which has been one of the earliest unifying literature analyses on the topic. In fact, the author is aware of no earlier works and only two works in this area submitted for publication marginally after the above-mentioned article [Rie+20; Nal+20], highlighting that gaining an overview of this field is much-needed scientific contribution. As of spring 2020, Riedmaier et al. noticed that "a comprehensive literature review, similar to the one included in the present paper, is not provided" [Rie+20].

Scenario-based testing provides methods for generating the data required for assessing SOTIF adherence. The next task is analyzing the data. Section 2.2.2 therefore presents two popular means of scenario data analysis: Criticality metrics and logics-based data querying. Criticality metrics provide a physical view on accident and severity proximity and can hence be used in assessing harm-related quantities for SOTIF estimation. The presentation is based on an extensive review by Westhofen et al. [Wes+23]. Published in 2023, it is, to the best of the author's knowledge, the first in-depth literature assessment on measuring criticality in data for automated driving applications. The second means for data analysis concerns quantities in traffic significantly preceding harm, which may trigger

hazardous behavior of the system. To allow querying for such environmental factors in road traffic data, logic-based approaches were introduced, for which we give an overview on the state-of-the-art.

The identified challenges in establishing SOTIF (cf. Section 2.1) together with the improved understanding of applying state-of-the-art scenario data analysis for SOTIF (cf. Section 2.2) leads to the specification of requirements on temporal querying from a practical perspective in Section 2.3. For example, while many logic-based approaches claim to be suitable for complex tasks such as traffic rule specification, none of these incorporate formal ontologies into the temporal specification. However, ontologies aid in unifying the extensive background knowledge provided by various stakeholders, such as legal experts and engineers [Wes+22b; SWH26]. These derived gaps form the basis of the query language presented and analyzed in the upcoming chapters.

2.1 Safety of the Intended Functionality

In December 2021, the first ADS gained type approval for operation on public German roads: The Mercedes-Benz Drive Pilot is an Automated Lane Keeping System (ALKS), a specific type of ADS with SAE Level 3 according to the taxonomy standard SAE J3016 [SAE21]. It takes over the driving task on highways under certain conditions, e.g., up to a speed of 95 kilometers per hour with a lead vehicle being present. In Germany, it has been approved on the basis of the United Nations Regulation No. 157 (UN157) [Uni21], which the system fully complies with according to the approval authority [Ste21]. This regulation was introduced in March 2021 and contains requirements on developing, releasing, and operating ALKSs [SAE21]. It was also adopted for the German Autonomous Vehicles Approval and Operation Regulation (AFGBV) [Ver22a], which, published in June 2022, additionally regulates the release of SAE Level 4 and 5 systems in Germany and implements the amendments of the German Road Traffic Act (StVG) in 2017 and 2021.

According to these regulations, a type approval of ADSs necessitates a demonstration of safety. Historically, the automotive industry considered mainly functional safety of manually driven vehicles. Functional safety as defined per ISO 26262 [Int18] concerns solely risks arising from non-compliance of items (systems with a sensor, controller, and actuator implementing a function at the vehicle level) to their specification. Activities within the scope of ISO 26262 contain, e.g., verification of the safety-critical software components against their requirements [WBK20]. When moving towards higher automation levels, a fallback driver cannot be assumed anymore. Hence, risks arising from the specification itself become more relevant – in case of unsafe specified behavior, no backup is available to take corrective action for guiding the system towards a safe state. For this, ISO 21448 proposes to complement ISO 26262 by considering the SOTIF, i.e., absence of

unreasonable risks arising from the specification and technical capabilities [Int22]. ISO 21448 has rapidly gained relevance and is also referenced in the above-mentioned UN157: "[Auditors/assessors] shall in particular be competent as auditor/assessor for [...] ISO 21448" [Uni21]. Substantiating this relevance, the German AFGBV also considers compliance with ISO 21448 as a sufficient criterion for adhering to the mandatory state-of-the-art safety assessment [Ver22a, Annex 1, Section 7.2]. ISO 21448 must therefore be viewed as a cornerstone in any safety activities around developing ADSs.

ISO 21448 was initiated in 2016, publicly available as a specification since 2019, and finally released in June 2022. As such, ISO 21448 compliant safety cases can now be conducted, which is, as delineated above, essential for compliance with the UN157 or the German AFGBV. In the end, it must be possible to devise a validation strategy implementing the standard's requirements. This necessitates a rigorous investigation of the ISO 21448. The work on which the subsequent elaboration is based on [Put+23] is, to the best knowledge, the first to perform such an investigation as its main topic.

2.1.1 ISO 21448 Safety Framework

As a first step, we examine the proposed terminological framework around safety according to ISO 21448. This allows comprehending the target, i.e., what achieving SOTIF means, as well as the artifacts created during the process. Despite the regulatory necessity of conducting SOTIF-compliant safety cases, the standard is unnecessarily inconsistent in its terminology, as noticed by Zhu et al. in 2022 [Zhu+22] and expanded upon by Putze et al. in 2023 [Put+23]. Therefore, before we dive into the proposed steps, we present a revised notation that resolves many inconsistencies. Among other issues, these result in the ill-definedness of the central term SOTIF in the ISO 21448.

The original terminological construct (of the most relevant terms) is given in Figure 2.1. If not otherwise defined in ISO 21448, the standard demands that the definitions of ISO 26262 apply [Int22, Clause 3]. Terms of ISO 26262 required for a complete definition of SOTIF are therefore included.

The central term is SOTIF, which is defined as "the absence of unreasonable risk due to hazards resulting from functional insufficiencies of the intended functionality or its implementation" [Int22, Definition 3.25]. Since SOTIF concerns only a subset of all hazards that can lead to an unreasonable risk of the final product, the chain of recursive definitions depicted in Figure 2.1 must be understood to fully comprehend above definition (terms defined in one of the two standards are *emphasized*):

- *Risk* combines the probability of occurrence and the *severity* of a *harm* [Int22, Definition 3.14].
  - *Harm* means inflicting damage to the health of persons [Int18, Definition 3.74].

  - *Severity* is an estimate of the extent of this damage in a *hazardous event* [Int18, Definition Definition 3.154].
    * A *hazardous event* is a combination of a *hazard* and an *operational situation* [Int18, Definition 3.77]. Additionally, the more general term *event* is defined by ISO 21448 as "occurrence at a point in time" [Int22, Definition 3.7].
      · A *hazard* is a possible cause of *harm* due to hazardous behavior exhibited by the vehicle [Int22, Definition 3.11], where the term hazardous behavior is not further defined.
      · An *operational situation* is a *scenario* encounterable by the vehicle [Int18, Definition 3.104].
        ○ While *scenario* is not further explicated in ISO 26262, ISO 21448 defines it as describing the temporal relations in a sequence of *scenes* (added goals and values lead to situations), where scenes are influenced by *actions* and *events* [Int22, Definition 3.26].
          ▷ A *scene* is representing the traffic at a fixed point in time [Int22, Definition 3.27].
          ▷ An *action* is a specific *event*: an executable behavior of an actor in a given *scene* [Int22, Definition 3.2].

- Risk is *unreasonable* if it is unacceptable by societal norms [Int18, Definition 3.176].
- *Functional insufficiencies* are divided into specification and performance insufficiencies [Int22, Definition 3.8]:
  - *Insufficiency of specification* is a specification. When activated by a *triggering condition*, this specification must contribute to a hazardous behavior or lead to failure in preventing or mitigating *reasonably foreseeable misuse* [Int22, Definition 3.12].
    * A *triggering condition* is a condition in a *scenario* that causes hazardous behavior or leads to failure in preventing or mitigating *reasonably foreseeable misuse* [Int22, Definition 3.30].
    * *Misuse* refers to an unintended usage [Int22, Definition 3.17].
    * While *reasonably foreseeable* is not defined in ISO 21448, ISO 26262 states that it means to be "technically possible and with a credible or measurable rate of occurrence" [Int18, Definition 3.120].
  - *Performance insufficiencies* are limitations of technical capabilities, with the same constraints as *insufficiency of specification* [Int22, Definition 3.22].

- The *intended functionality* is simply equivalent to the specified functionality [Int22, Definition 3.14].

Note that "implementation" is left undefined, but most likely acts as the subject of performance insufficiencies. Despite this complex network of definitions, a possible interpretation of above terminology is that ISO 21448 proposes to ensure that the probability and extent of damage to persons caused by the vehicle's behavior and originating from the specification or the implementation's technical capabilities are socially acceptable. This is in contrast to the topic of ISO 26262, where risk is only considered when it originates from malfunctioning behavior, i.e., where the system or its components fail to behave like intended.

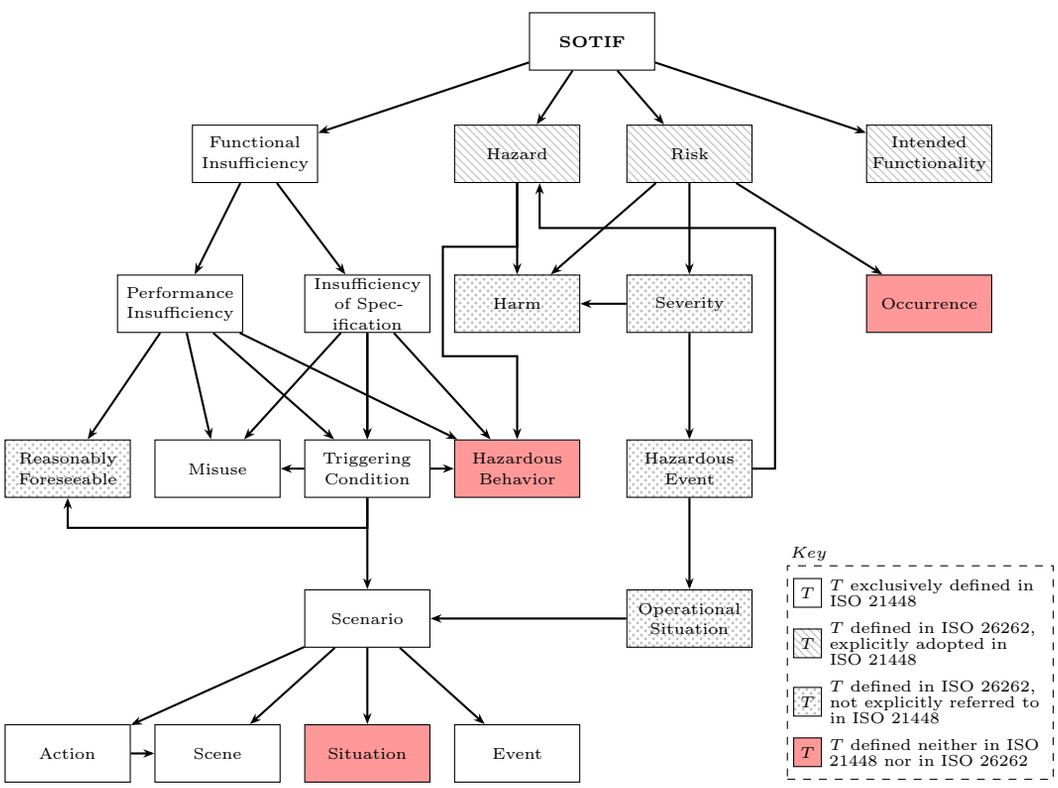

Figure 2.1: The terms used in the risk framework of ISO 21448, together with their implicit references to ISO 26262, based on Putze et al. [Put+23]. If a term uses another term in its definition, an arrow is drawn in that direction.

First, it is easy to note that this chain of definition is excessively complex, unnecessarily obstructing the understanding of the central objective of ISO 21448. Second, several technical issues arise from the terminology and are most likely caused by the high complexity and implicit dependencies on other standards:

1. The definition of *harm* explicitly excludes damage to property and is in violation with the meta-standard ISO/IEC Guide 51 [Int14], which gives guidance and requirements for risk standard developers. Even if this incompatibility with ISO/IEC Guide 51 has been a deliberate decision, no note with a rationale is attached.

2. *Hazardous behavior* is frequently used in above definitions (hazard, insufficiency of specification, and triggering condition) but remains undefined. The most likely reason is that giving a definition in above the framework would lead to circular dependencies. Based on Figure 13 in ISO 21448, it is causally situated between triggering conditions and hazards. Yet, it cannot be based on the definition of hazard, e.g., defining it as "a behavior of the system leading to a hazard," since hazard itself is defined based on hazardous behavior. Similarly, it cannot be defined as a cause of some triggering condition or functional insufficiency, as also triggering condition is defined using the term hazardous behavior.

3. For references to *hazardous event* in ISO 21448, it is unclear whether the definition of hazardous event from ISO 26262 or the definition of event from ISO 21448 shall be applied. Both are inconsistent with each other: While ISO 26262 refers to hazardous event as a scenario (where scenario is not further explicated in ISO 26262 but becomes defined as a temporal entity in ISO 21448), ISO 21448 views an event simply as a point in time. These conflicting views leave the term hazardous event unclear within ISO 21448. Since hazardous event is required to understand severity and thereby risk as well as SOTIF, the implications are far-reaching. If we assume that the more specific definition of ISO 26262 shall be applied, several terms used in its definition break with the scope of ISO 21448. This includes hazard, which is, for ISO 26262, restricted to malfunctioning behavior. Yet, malfunctioning behavior is out of scope for ISO 21448. If we thus apply the definition of ISO 26262 of hazardous event, any ISO 21448-compliant severity analysis must only consider malfunctioning behavior and completely disregard unsafe specifications, which breaks with the scope of SOTIF.

4. The definition of *scenario* uses the term situation, which is undefined in ISO 21448 – most likely to avoid conflicts with the term operational situation from ISO 26262: An operational situation must be a specific situation, by naming, and also a scenario, by definition. Any definition of the term situation that does not allow certain scenarios to also be situations would violate the above constraint. However, scenarios are defined using situations, which makes it hard to avoid circular reasoning when trying to incorporate scenarios into the definition of situation. On top of this issue, the term does not occur in the referenced original definition by Ulbrich et al. [Ulb+15], and the trouble could have been avoided had ISO 21448 simply included the verbatim original definition.

5. *Intended functionality* is equivalent to the specified functionality, and therefore the question arises why a distinct term is even necessary. In fact, the specifiers' intention might not even match the specification, and therefore the term intended functionality can be misleading. This might raise false expectations regarding the scope of ISO 21448.

These issues propagate transitively through the complex definition tree depicted in Figure 2.1, leading to ill-definedness of the central term SOTIF.

The following definitions, mostly taken from Putze et al. [Put+23], remove the above-mentioned issues and significantly improve understandability (however, they do not resolve the issue of having an unnecessarily convoluted definition chain). First, we include property damage in harm to comply with ISO 51.

Definition 2.1.1 (Harm [Int14])

"Injury or damage to the health of people, or damage to property or the environment."

In order to be able to define hazardous behavior, we require removing a reference to hazardous behavior from either triggering condition or hazard. We opt for the latter, as ISO 51 already gives a suitable definition without reference to hazardous behavior.

Definition 2.1.2 (Hazard [Int14])

"Potential source of harm."

Then, hazardous behavior can be defined without circular dependencies.

Definition 2.1.3 (Hazardous Behavior [Put+23])

"Behavior of the functionality which can lead to a hazard."

Regarding hazardous events, we delineated how implicitly relying on terms from ISO 26262 can destruct large parts of the framework built in ISO 21448 as the considered categorizations of hazard are different. Hence, the proposal is to explicitly rely only on definitions from ISO 21448 or our updated terminology:

Definition 2.1.4 (Hazardous Event [Put+23])

"Event [Int22, Definition 3.7] that is a combination of a hazard and a scenario containing conditions in which the hazard can lead to harm."

Similar issues are observable with the term scenario, where an implicit dependency on the operational situation leads to inconsistencies. Therefore, we simply propose to rely on the original definition of Ulbrich et al. [Ulb+15].

Definition 2.1.5 (Scenario [Ulb+15])

"A scenario describes the temporal development between several scenes in a sequence of scenes. Every scenario starts with an initial scene. Actions and events as well as goals and values may be specified to characterize this temporal development in a scenario. Other than a scene, a scenario spans a certain amount of time."

We remark that several qualifications of the term scenario – most importantly, concrete, logical, abstract, and functional – exist, which are used in the literature without clear semantics. For further reference, the reader is referred to Neurohr et al., who provide a formal framework for scenarios and define the qualifications [Neu+26].

The final issue concerns the term intended functionality, which is identified with specified functionality. Since removing the term altogether means changing the name of the standard itself, the easier (and more complete) solution is to simply include mismatches between intention and actual specification to the scope of the standard.

Definition 2.1.6 (Intended Functionality)

"Functionality intended at time of specification."

By this terminology, it becomes clear that SOTIF is concerned with ensuring societal acceptance of the probability and extent of any damage caused by the vehicle's behavior which originates from the intended or written specification, or the implementation's technical capabilities. Moreover, the causal model underlying harm is now more precise.

Example 2.1.1

An example of the SOTIF causal model from functional insufficiencies to harm is given in Figure 2.2. Here, two functional insufficiencies – one performance insufficiency and one insufficiency of specification – can lead, together with a corresponding triggering condition, to hazardous behaviors of the vehicle. The first insufficiency concerns the technical capabilities of the employed sensors: A truck driving in front of the vehicle is not recognized due to ice plates being on top, e.g., the image classifier was not trained on such data, leading to an adjusted distance to the truck. The second insufficiency arises due to an insufficient incorporation of falling loads into the specification. If the ice plates fall onto the road, a wrong evasive maneuver can be hazardous. According to Definition 2.1.3, hazards – i.e., possible causes of damage to the occupants, others, or property – can now arise from this behavior. The first hazardous behavior (unadjusted distance) must become inappropriate to be able to lead to harm. For example, the distance is too close to evade falling ice plates, which forms the subsequent hazardous event. The second hazardous behavior (evasive maneuver) can become a hazard if it is applied inappropriately, e.g., if some vehicle is one the other lane. Alternatively, it can lead to swerving if the lateral jerk

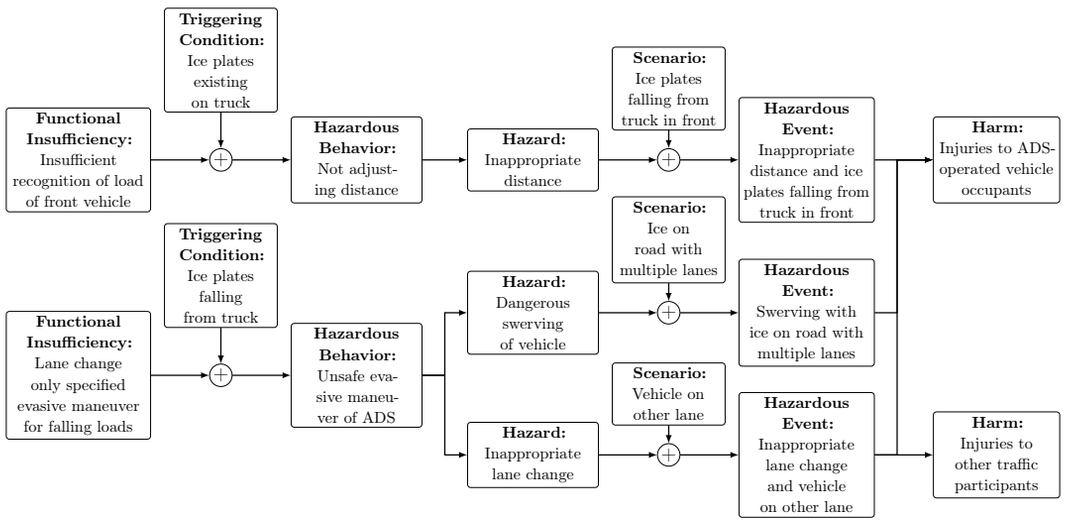

Figure 2.2: Example of causal factors underlying several harms, adapted from Putze et al. [Put+23].

is too strong, which might in turn lead to the vehicle driving into the oncoming traffic or rolling over.

Several conclusions can be drawn from this example:

1. Omission of behavior can also be a hazardous behavior itself, such as not adjusting longitudinal distance.

2. There is no one-to-one relation between the elements of the causal model preceding harm. For example, multiple functional insufficiencies and triggering conditions can lead to the same hazardous behavior, which in turn can split up to into multiple hazards only to be rejoined again into a single harm. Hence, it can only be assumed that such a causal model is a directed acyclic graph (cycles would imply jumping back in time).

3. Boundaries between elements in the causal model are fluid. In fact, Figure 2.2 shows that falling ice plates can both be a scenario and a triggering condition. Putting a causal factor in one of the categories is an engineering decision, as the next issue shows distinctly.

4. It is often of little use to declare hazards as collisions (contrary to the example given in Figure 13 of ISO 21448 [Int22]). When assessing the behavioral safety of ADSs, we are interested in finding effective mitigation strategies. Hence, it makes little sense to locate hazards at the very end of the causal model as otherwise, countermeasures simply collapse to “avoid collisions” [Cle00, Sheet 84-3].

This example shows how causes of harm are typically decomposed in ADS safety analyses. The overall goal is twofold: Engineers can improve the specification to avoid or mitigate the impact of these causes. For this reason, the causal analysis requires traversing the chain of events backwards up the system specification or capability level, i.e., functional insufficiencies. Additionally, decomposition eases risk quantification, which is, in many areas, imperative for approval. Often, the likelihood of harm occurring is impossible to estimate directly, and therefore it might be easier to combine an estimate of the occurrence of its causes with an estimated harm in such hazardous events. The concept of quantifying a risk decomposition is now further examined.

2.1.2 ISO 21448 Risk Quantification

Based on this revised and now well-defined safety framework, we can assess how ISO 21448 proposes to evaluate the achievement of SOTIF. As a contribution, we present a formula derived from the work of Putze et al. for a SOTIF-compliant risk decomposition [Put+23]. It significantly improves upon what is proposed in Annex C.2 of ISO 21448 in several key aspects and hence advances the implementability of ISO 21448-compliant safety cases. As we will later see, scenario-based data analysis (including the tools proposed in this dissertation) can aid in estimating the required quantities of this formula.

While ISO 21448 provides many process steps to increase chances of meeting SOTIF (such as identifying relevant hazards in Clause 6 and modifying the functionality in Clause 8), one of the main concerns is demonstrating that SOTIF actually holds. For this, ISO 21448 proposes to rely on so-called acceptance criteria derived from the steps of Clause 6 and attached to the collected hazardous behaviors. If fulfilled, acceptance criteria prove that unreasonable risk is absent for a given hazard [Int22, Definition 3.1]. In the verification and validation activities of Clauses 9 and 10 it is checked whether these criteria are actually met by the to-be-deployed system. This is done by deriving, for each acceptance criterion, a validation target specifying a measurable quantity of evidence required for confidently arguing that the acceptance criterion holds. Annex C.2 provides an example of how an acceptance criterion A_H for a harm H might be decomposed in order to later derive a validation target for a fixed hazardous behavior HB:

$$A_H = R_{HB} \cdot P_{E|HB} \cdot P_{C|E} \cdot P_{S|C}, \tag{2.1}$$

taken verbatim from the standard [Int22, Annex C.2.1], where R_{HB} is the rate of the hazardous behavior, $P_{E|HB}$ is the exposure, i.e., the probability of the hazardous behavior leading to a hazardous event, $P_{C|E}$ is the controllability, i.e., the probability of the hazardous event leading to a harm, and $P_{S|C}$ is the severity, i.e., the probability that harm has a certain extent. Here, the idea is to compute the acceptable rate of hazardous behavior R_{HB} based on some previously socially agreed upon upper bound H_{max} on the rate of harm A_H, e.g., $A_H \leq H_{max} = 10^{-8}$ per operating hour, and estimations of

the quantities $P_{E|HB}$, $P_{C|E}$, and $P_{S|C}$. This rate R_{HB} can then be used for deriving a validation target, which tells test engineers the to-be-covered distance in which no hazardous behavior should occur [Int22, Annex C.3].

While the general idea is promising, several technical issues exist with above formula:

1. It uses unconventional probabilistic notation. Specifically, it omits existing conditions. For example, $P_{C|E}$ must also be conditioned on the hazardous behavior, i.e., applying the same notation, it should be formulated as $P_{C|E,HB}$.

2. It assumes a one-to-one relation between a single harm, a single hazardous event, and a single hazardous behavior, which violates the second conclusion drawn from the example of Figure 2.2. Rather, if the harm is fixed, an aggregation over *all* hazardous events and behaviors leading to this harm must be performed, taking into account that these might not be independent events. This assumption is not explicitly mentioned in Annex C.2.1 of ISO 21448.

3. It does not consider the quantities $P_{E|HB}$, $P_{C|E}$, and $P_{S|C}$ to be system-dependent. On the contrary, the standard mentions in an example that they can be taken "from field data." However, a well-founded estimation is impossible without running the actual system, as even exposure to scenarios depends on strategic routing decisions and the spatial and temporal distribution of origin and the destination of users. If a prior estimation on these quantities is performed, assumptions on these strategic variables must be articulated explicitly.

Putze et al. therefore present a revised version of Equation (2.1), which fixes these issues and enables us to identify which kind of data analysis is required for estimating SOTIF compliance [Put+23, Equation 3]. Assume the set of harms is $\mathfrak{H}$. For $H \in \mathfrak{H}$, we define $\mathfrak{E}^H$, $\mathfrak{B}^H$, and $\mathfrak{T}^H$ as the sets of all known hazardous events, hazardous behaviors, and triggering conditions that are linked to H. For each $H \in \mathfrak{H}$, $E \in \mathfrak{E}^H$, $B \in \mathfrak{B}^H$, and $T \in \mathfrak{T}^H$ the corresponding events in the probability space are denoted by $\mathcal{H}_H$, $\mathcal{E}_E^H$, $\mathcal{B}_B^H$, and $\mathcal{T}_T^H$, respectively. These events are all dependent on the system's behavior. Assuming completeness of the underlying causal model, i.e., no factor leading to H was missed, a bound on the probability of H with severity $\mathcal{S}$ of a given level S can be given as [Put+23]:

$$P(\mathcal{H}_H, \mathcal{S} = S) \leq \sum_{\substack{E \in \mathfrak{E}^H \\ B \in \mathfrak{B}^H \\ T \in \mathfrak{T}^H}} \begin{array}{l} P(\mathcal{T}_T^H) P(\mathcal{B}_B^H | \mathcal{T}_T^H) P(\mathcal{E}_E^H | \mathcal{B}_B^H, \mathcal{T}_T^H) \\ P(\mathcal{H}_H | \mathcal{E}_E^H, \mathcal{B}_B^H, \mathcal{T}_T^H) P(\mathcal{S} = S | \mathcal{H}_H, \mathcal{E}_E^H, \mathcal{B}_B^H, \mathcal{T}_T^H) \end{array} \tag{2.2}$$

This improves the issues identified in Equation (2.1). Note that an equality can be achieved by incorporating the inclusion-exclusion principle. However, we omit these terms, as estimating probabilities for the intersection of events adds to the already high practical burden. Additionally, other decompositions of risk are possible, e.g., by using Bayes'

theorem to avoid estimating some of the quantities above [Bab+23, Equation 1]. However, this section is solely concerned with improving the suggested Equation (2.1) of ISO 21448.

Annex C.2 continues with deriving a bound on the tolerable rate of a fixed hazardous behavior that would satisfy previously agreed upon upper bound H_{max} on the rate of harm of a given severity. We revise this line of thought using our updated framework, and thereby extending the work of Putze et al. [Put+23].

Two main issues arise from Annex C.2 in ISO 21448. First, the presented calculations assume an equality, as given in Equation (2.1). Our updated Equation (2.2) uses instead an inequality, due to the correct incorporation of the causal network preceding harm and the impracticability of the inclusion-exclusion principle. We will hence show how SOTIF can be proven under this inequality. Second, Equation (2.1) does not explicitly mention any triggering condition preceding the fixed hazardous behavior. Yet, the idea behind SOTIF is to analyze the hazardous behavior based on the occurrence of its triggering conditions (as suggested by the scenario-based approach presented in Section 2.2.1). We therefore add a fixed triggering condition $T \in \mathfrak{T}^H$ preceding the hazardous behavior under analysis $B \in \mathfrak{B}^H$. This allows being compliant with the proposed procedure of Annex C.2, i.e., fixing $B \in \mathfrak{B}^H$, while avoiding estimating the unconditional probability of the hazardous behavior directly.

We now address both issues. Achieving SOTIF formally means that $\mathit{Risk}(S, H) < H_{max}$. Our goal is to derive a bound on the hazardous behavior that suffices to show that SOTIF holds. For ease of presentation, we define the following abbreviations to be used in what follows:

- $\mathit{Risk}(S, H) \coloneqq P(\mathcal{H}_H, \mathcal{S} = S)$,
- $\mathit{Behavior}(H, B, T) \coloneqq P(\mathcal{B}_B^H | \mathcal{T}_T^H)$, and
- $\mathit{Remainder}(S, H, B, T) \coloneqq P(\mathcal{T}_T^H) \sum_{E \in \mathfrak{E}^H} P(\mathcal{E}_E^H | \mathcal{B}_B^H, \mathcal{T}_T^H) P(\mathcal{H}_H | \mathcal{E}_E^H, \mathcal{B}_B^H, \mathcal{T}_T^H)$
  $P(\mathcal{S} = S | \mathcal{H}_H, \mathcal{E}_E^H, \mathcal{B}_B^H, \mathcal{T}_T^H)$.

Note that $\mathit{Remainder}(S, H, B, T)$ has an estimable value, whereas the probability of hazardous behavior is still unknown – we aim to identify an upper bound for this quantity. A sufficient criterion for SOTIF can be derived using the following steps:

1. SOTIF is violated iff $H_{max} \leq \mathit{Risk}(S, H)$. From Equation (2.2), it follows that $\mathit{Risk}(S, H) \leq \mathit{Behavior}(H, B, T) \mathit{Remainder}(S, H, E, B, T)$.
2. This allows eliminating the term $\mathit{Risk}(S, H)$, which is desired because determination of this quantity is deemed infeasible in practice [KP16; WW16]. Hence, violation of SOTIF implies $H_{max} \leq \mathit{Behavior}(H, B, T)\ \mathit{Remainder}(S, H, E, B, T)$.
3. By contraposition, $H_{max} > \mathit{Behavior}(H, B, T) \mathit{Remainder}(S, H, E, B, T)$ implies the satisfaction of SOTIF.

Since, in practice, we can estimate the probability of triggering conditions to a sufficient degree, we can rearrange further to achieve

$$Behavior(H, B, T) < \frac{H_{max}}{Remainder(S, H, B, T)}. \tag{2.3}$$

In contrast to what is presented in ISO 21448, we now have only a *sufficient* criterion for SOTIF. It is thus a valid strategy for test engineers to demonstrate the satisfaction of the bound on hazardous behavior (possibly under a predefined confidence level). In terms of ISO 21448, Equation (2.3) is an acceptance criterion [Int22, Definition 3.1]. However, it might also happen that the risk bound of Equation (2.1) is not tight enough for satisfying the above sufficient condition – even if SOTIF actually holds. Then, an exact value for Equation (2.2) must be computed using the inclusion-exclusion principle, resulting in an equivalence between achieving SOTIF and Equation (2.3).

Example 2.1.2

Consider the harm "injuries to ADS-operated vehicle occupants" (*Occ*) from Figure 2.2 under a high severity, which we define as a score of 5 or 6 on the Maximum Abbreviated Injury Scale (MAIS). MAIS is defined as the largest classified injury according to the Abbreviated Injury Scale (AIS), where the AIS assigns the score 5 to "critical" and 6 to "virtually unsurvivable" injuries [GMG85]. For simplicity, this example uses a harm with only a single identified underlying causal chain.

As a first step, a bound on the probability of encountering the harm at the given severity, i.e., the risk, must be agreed upon. In the example, consider it is desired to achieve $\hat{P}(\mathcal{H}_H, \mathcal{S} = S) \leq 10^{-9} = H_{max}$. Since the probabilities of the events highly depend on the ODD and system characteristics, our example analyzes a passenger car equipped with a highway ALKS in the German city of Berlin. In order to derive a validation target according to Annex C.3 of ISO 21448, the following estimates were obtained.

We start with the probability of encountering the triggering condition "ice plates falling from truck in front" (*Ice*). As a rough approximation, consider that Berlin had an average of 39.4 snowy days per year between 2000 and 2024[1]. Assuming the system drives under arbitrary weather conditions and always chooses the rightmost lane on the highway, it yet remains to identify the probability of encountering a truck whose cargo was not cleared from snow. While the probability of uncleared ice or snow on truck cargo is hard to estimate, an e-mail-based survey of German terminal and trucking companies shows that appropriate means to remove ice and snow from cargo are often missing, and 40% of the companies responded to not remove snow or ice [Stu15]. For the example, we assume that 50% is a conservative upper bound. Since the system drives on the rightmost lane, the probability of having a truck as a lead vehicle is also considered to be roughly 50%.

[1] According to `weatheronline.de`, as accessed on 3rd of September 2025.

Finally, the triggering conditions requires the ice to be slipping of, which we, conservatively, estimate as 50% as well. Therefore, an estimation of the probability of encountering the triggering condition might be given as $\hat{P}(\mathcal{T}_{Ice}^{Occ}) = \frac{39.4}{365.25} \cdot 0.5^3 \approx 1.35 \cdot 10^{-2}$. Note that a more specific behavior of the system influences this estimation, e.g., if the ALKS mostly avoids driving on the rightmost lane, $\hat{P}(\mathcal{T}_{Ice})$ will be significantly lower.

Since we later aim to check whether the estimated probability of an unsafe evasive maneuver (the hazardous behavior), $\hat{P}(\mathcal{B}_{Evade}^{Occ}|\mathcal{T}_{Ice}^{Occ})$, is low enough to comply with the identified bound on harm, its quantification is postponed to a downstream process step.

The hazardous event "swerving with ice on road with multiple lanes" (*Swerve*) happens, among the above scenarios, only in a few cases, as we use a modern vehicle equipped with an electronic stability control system. Extensive tests in simulation and on specifically prepared proving grounds show this event to only take place with a rate of 10^{-4}. Furthermore, the road must be icy, which happens mostly at night and early morning hours of snowy days, as the road is typically gritted in the morning, i.e., the time slot for encountering icy roads is roughly 20% of the day. Taking into account a ten times less likely operation of the ALKS at those times, the system encounters icy roads during snowy days occur at a rate of 0.02. Overall, $\hat{P}(\mathcal{E}_{Swerve}^{Occ}|\mathcal{B}_{Evade}^{Occ}, \mathcal{T}_{Ice}^{Occ})$ is given as $2 \cdot 10^{-6}$.

The probability of the ADS occupants obtaining *any* injury when rolling over in highway scenarios under is conservatively estimated as almost certain, and we define $\hat{P}(\mathcal{H}_{Occ}|\mathcal{E}_{Swerve}^{Occ}, \mathcal{B}_{Evade}^{Occ}, \mathcal{T}_{Ice}^{Occ}) = 1$.

Finally, the chances of obtaining a score of 5 or 6 on the MAIS in a rollover accident on German highways are roughly 5.1%, based on representative data from the German In-Depth Accident Study (GIDAS) between 1994 and 2000 [OK05, Figure 5], and therefore we put $\hat{P}(\mathcal{S} = \mathit{High}|\mathcal{H}_{Occ}, \mathcal{E}_{Swerve}^{Occ}, \mathcal{B}_{Evade}^{Occ}, \mathcal{T}_{Ice}^{Occ}) = 0.051$.

The probabilities are summarized in Table 2.1. Note that this example is intentionally incomplete: Many other causes can injure occupants as well, and hence the actual risk is much higher.

Table 2.1: Estimated probabilities for the example harm "injuries to ADS-operated vehicle occupants" for a severity score of 5 or 6 on the MAIS and the causal model of Figure 2.2.

Quantity	Value
$\hat{P}(\mathcal{T}_{Ice}^{Occ})$	$2.7 \cdot 10^{-2}$
$\hat{P}(\mathcal{B}_{Evade}^{Occ}\|\mathcal{T}_{Ice}^{Occ})$	Not yet estimated
$\hat{P}(\mathcal{E}_{Swerve}^{Occ}\|\mathcal{B}_{Evade}^{Occ}, \mathcal{T}_{Ice}^{Occ})$	$2 \cdot 10^{-6}$
$\hat{P}(\mathcal{H}_{Occ}\|\mathcal{E}_{Swerve}^{Occ}, \mathcal{B}_{Evade}^{Occ}, \mathcal{T}_{Ice}^{Occ})$	1
$\hat{P}(\mathcal{S} = \mathit{High}\|\mathcal{H}_{Occ}, \mathcal{E}_{Swerve}^{Occ}, \mathcal{B}_{Evade}^{Occ}, \mathcal{T}_{Ice}^{Occ})$	$5.1 \cdot 10^{-2}$
H_{max}	10^{-9}

Continuing the example, we now derive an acceptable bound on the probability of hazardous behavior as per Equation (2.3). Specifically, SOTIF can be considered achieved if

$$Behavior(Occ, Evade, Ice) < \frac{10^{-9}}{2.7 \cdot 10^{-2} \cdot 2 \cdot 10^{-6} \cdot 1 \cdot 5.1 \cdot 10^{-2}} \approx 36.31 \cdot 10^{-2}.$$

If an unsafe evasive maneuver occurs in at most 36.31% of the cases where ice plates are falling from a lead truck, SOTIF holds.

Example 2.1.2 assumed the probability of harm given the hazardous event to be validly estimated. One of the reasons behind selecting a conservative upper bound is that estimating the probability of harm in such events is often hard: We require data where the hazardous event was observed many times, and sufficiently many cases of the hazardous event lead to injury or damage. This can only be obtained using simulated or real accident data. When relying on the latter, it is not always guaranteed that it is possible to identify the hazardous event a-posterior based on the data collected from on-site personnel – in an example study based on the GIDAS, only 70% of a large collection of critical events were identifiable based on the collected data and the database scheme [Bab+23, p. 10]. Using simulation, on the other hand, requires a large amount of samples obtained from highly valid behavior and collision models of the involved actors, which can make SOTIF assessment economically infeasible.

A possible solution is to employ proxies for measuring closeness to accidents of a certain severity – such as violating the safety distance with a high relative velocity – instead of measuring harm directly. Accident proxy exceedance occurs at higher rates and therefore can be used to either increase confidence or allow lowering the burden on data collection. Additionally, defining how many injuries or deaths are acceptable while the system is in operation can be a daunting task. Consider that we simply put 10^{-9} as a bound in Example 2.1.2 – yet, why not put it at 10^{-10} or even 10^{-20}? The decision inevitably leads to evaluating the worth of human life on more or less subjective scales.

Agreeing upon an acceptable level of exceeding some accident proxy might be viewed as a more agreeable option. If we assume that the exceedance of a valid accident and severity proxy for a harm H is indicated by an event $\mathcal{A}_H$, we can change Equation (2.2) to

$$P(\mathcal{A}_H) \leq \sum_{E \in \mathfrak{E}^H, B \in \mathfrak{B}^H, T \in \mathfrak{T}^H} P(\mathcal{T}_T^H) P(\mathcal{B}_B^H | \mathcal{T}_T^H) P(\mathcal{E}_E^H | \mathcal{B}_B^H, \mathcal{T}_T^H) P(\mathcal{A}_H | \mathcal{E}_E^H, \mathcal{B}_B^H, \mathcal{T}_T^H) \tag{2.4}$$

Note that the approach requires arguing for the accuracy of the selected accident proxies, a topic we will further study when presenting criticality metrics in depth. Equation (2.4) can now be used for downstream computation of validation targets for hazardous behaviors, analogously to what has been done to derive Equation (2.3).

Example 2.1.3

We continue Example 2.1.2. Instead of measuring harm directly, we now use an indicator proxy I. In this case, we rely on Time Headway (THW) – measuring the time until the current position of the truck is reached, assuming constant velocity, i.e., $\frac{d}{v}$ – in combination with a reciprocal speed factor to account for potential severity:

$$I(v, d) = \frac{d}{v} \cdot \frac{1}{v} = \frac{d}{v^2}$$

Here, v denotes the vehicle speed and d its distance to the leading truck. An argument can be made that THW offers a high sensitivity [Wes+23, Section 5.2.10] due to its use in legislation such as the German Road Traffic Regulation (StVO) [Ver13, §4 (1)]. Speed, obviously, is a proxy for the amount of energy that might need to be absorbed by the car, its occupants, or other objects in the event of a potential collision. Additionally, we require a sound target value below which this indicator considers the event to be critical. For simplicity, we rely on the typical threshold of 1.8 seconds for THW [Wes+23] and calibrate it using a reference speed of 25 meter per second, leading to a target value of 0.072 seconds squared. Note that this value can also depend on the specifics of the scenario, e.g., for THW, it was found that correlation of THW to (subjective) risk depends on speed [RFA18]. As the final component, a discussion between stakeholders reveals that an acceptable bound on the rate of violating $I(v, d) \geq 0.072$ is 10^{-7}.

Table 2.2: Estimated probabilities for the example harm "injuries to ADS-operated vehicle occupants" of Example 2.1.2 under an indicator approximating accident probability and severity.

Estimated Probability	Value
$\hat{P}(\mathcal{T}_{Ice}^{Occ})$	$2.7 \cdot 10^{-2}$
$\hat{P}(\mathcal{B}_{Evade}^{Occ} \mid \mathcal{T}_{Ice}^{Occ})$	Not yet estimated
$\hat{P}(\mathcal{E}_{Swerve}^{Occ} \mid \mathcal{B}_{Evade}^{Occ}, \mathcal{T}_{Ice}^{Occ})$	$2 \cdot 10^{-6}$
$\hat{P}(\mathcal{A}_{Occ} \mid \mathcal{E}_{Swerve}^{Occ}, \mathcal{B}_{Evade}^{Occ}, \mathcal{T}_{Ice}^{Occ})$	0.5
Acceptable bound on $\hat{P}(\mathcal{A}_{Occ})$	10^{-7}

We now execute the ADS in a simulation representing the hazardous event and record the proxy event $\mathcal{A}_{Occ}$ as present if $I(v, d) < 0.072$. Note that we removed the burden of incorporating highly valid collision models in the simulation. We find that in these events, the proxy is violated in 50% of the cases. All probabilities are given in Table 2.2, where the remaining values are taken from Example 2.1.2. Applying the same procedure as in Example 2.1.2 to these values, we obtain an acceptable probability of hazardous behavior in case of ice plates falling from a truck in front of 37.04%.

To conclude, evaluating SOTIF achievement by estimating $P(\mathcal{H}_H, \mathcal{S} = S)$ directly is deemed uneconomical [KP16; WW16]. Hence, ISO 21448 proposes a decompositional approach based on hazards, which we revised both in terminology and quantification aspects. Based on the resulting Equation (2.2), we additionally derived Equation (2.4), which can allow for an easier estimation of some involved components. If values for all of them can be estimated, we can derive a bound on the rate of hazardous behavior, to which adherence is then demonstrated during subsequent verification and validation. While the given examples already gave hints on how to estimate the involved quantities, a more thorough investigation is conducted in the next section.

2.2 Evaluating SOTIF: Selected State-of-the-Art

We have seen that one possible solution to assess SOTIF is estimating Equation (2.4) based on data. Two steps are required for this: Methods for efficiently generating data to estimate its components as well as measuring the presence of the events in the generated data. We give an overview on the first issue in Section 2.2.1 and will take a deeper look at the second component in Section 2.2.2.

2.2.1 Scenario-Based Testing for SOTIF Evaluation

In the examples given in Section 2.1, all quantities of Equation (2.4) require estimation, except for the probability of the hazardous behavior (which must also be estimated, but only during the subsequent verification and validation phase of ISO 21448). Recall that all components of this formula are generally system-dependent, and therefore it must either be argued why the system has little to no influence on the quantity, or the system has to be incorporated while gathering data. Whereas the first option can certainly be implemented in special circumstances, it is not generally applicable. Hence, *testing* plays an important role in estimating said quantities, where testing is the "process of planning, preparing, and operating or exercising an item or element to verify that it satisfies specified requirements, to detect safety anomalies, to validate that requirements are suitable in the given context, and to create confidence in its behavior" [Int18]. As ISO 21448 uses scenarios to represent inputs to the system, for estimating the required quantities, we can make use of the techniques developed in the field of *scenario-based testing*. The following presents a brief presentation of state-of-the-art and accompanying challenges when using scenario-based testing for estimating the quantities of Equation (2.4), which is loosely based on the works of Neurohr et al. [Neu+20; Neu+26].

Scenarios The central subject of scenario-based testing are *scenarios*, as indicated by its name. They are also a foundational concept in ISO 21448 [Int22, Definition 3.26]. A widely acknowledged definition has been given by Ulbrich et al. in 2015 [Ulb+15],

who recognized the need for unifying the semantics of scenario and its constituents. Their conceptualization was already presented in Definition 2.1.5. Intuitively, a scenario describes the temporal relation of scenes, which in turn are snapshots of the traffic environment. We remark that there are two ways of how scenarios can arise in practice: First, by recording long-term data, e.g., by observing an intersection with a drone [Boc+20], and subsequently divide these into smaller chunks, each then representing a scenario. The second possibility is to define or constrain the shape of the scenario by some specification language and use a data-generating process, such as a simulator or proving ground setup, for instantiating the specification. We now take a closer look at the second category, since understanding the employed scenario specification languages allows for insights into the role of scenarios within scenario-based testing.

Historically, the first accounts of using languages to describe the temporal evolution of traffic happenings for traffic simulations can be traced back to at least the 1990s. Here, Kearny et al. have used scenarios to describe placement and behavior of objects surrounding a vehicle used for driver training or studies [Kea+99]. Introduced in 2005, SILAB similarly focuses on a scenario-based simulation in the same context [Kru+05]. With regard to examining driver assistance systems, the first written accounts date back to 2007, where the software tool Virtual Test Drive (VTD) specified its inputs in the form of "test scenarios" [NDW09]. Prior to that, in 2006, the domain `openscenario.org` was registered. First discussions on the scenario language OpenSCENARIO XML has taken place that year in the context of the OpenDRIVE language used for specifying road networks [Gmb16]. The first OpenSCENARIO meeting has been held in October 2015, with the eventual goal of achieving industry consensus towards standardization. The resulting XML-based scenario description language for imperatively specifying placement and behavior of actors has been continuously developed under the umbrella of Vires, the company behind VTD, using an open-contribution model. OpenSCENARIO XML has gained traction during the PEGASUS and ENABLE-S3 projects between 2016 and 2019. The format has been handed over to the Association for Standardization of Automation and Measuring Systems (ASAM) in 2020, leading to the first ASAM release of the OpenSCENARIO XML standard.

Parallel to using scenarios as concrete step-by-step instructions for simulators, several logics for specifying finite-time sequences of traffic happenings were proposed, with the purpose of safety analysis of driving automation features. These include the Multi-Lane Spatial Logic in 2011 [Hil+11], Visual Logic in 2013 [KE13], and Traffic Sequence Charts in 2017 [Dam+17]. Scenarios in this category are also called *abstract scenarios* [Neu+21], which use a declarative, logics-based approach, in contrast to the imperative style of OpenSCENARIO XML [ASA24d], and include most prominently OpenSCENARIO DSL [ASA24c], being based, among others, on a domain specific language used within CARIAD [Boc+19] and the Measurable Scenario Description Language by Foretellix [Ltd20]. To

correspondingly widen the scope of these imperative specifications as well, these so-called *concrete scenario* languages were extended towards *logical scenarios* [MBM18], which additionally allow for parameters in the scenario description (such as taking speeds from a given interval). For example, OpenSCENARIO XML has added parameter functionality in 2021 [ASA21]. We refer to Neurohr et al. for a detailed account of the historical developments in scenario descriptions as well as a formal investigation of the differences between the above scenario qualifications [Neu+26].

Testing Process As already suggested at above, there are two possible paths to perform scenario-based testing:

- *implicitly*, by continuously operating the system in its environment and creating scenarios by cutting the recording into chunks, and
- *explicitly*, by first specifying scenarios and then operating the system solely in that scenario.

Regarding the first strategy, we briefly remark that possible ways of continuously operating the ADS in its environment include real-world test drives, e.g., with a qualified test driver or in public beta phases without on-scene supervision [Kus+25], and simulating traffic of a complete road network, such as the diverse maps available in some nanoscopic traffic simulators like Carla [Dos+17], or by coupling it with a microscopic road network simulation like SUMO [Kra+02]. These implicit techniques of operating the system in its context have the advantage of potentially greater validity while not requiring an explicit enumeration of all possible inputs prior to execution. On the other hand, this implicitness often comes at the expense of higher economic burdens, as it lacks effective strategies to guide the system into critical input spaces, and testing duration increases.

Explicit scenario-based testing on the other hand allows for a more directed operation of the system in its input space. An abstract workflow is shown in Figure 2.3, adapted from Neurohr et al. [Neu+20, Figure 1] to fit in the scope of SOTIF. First, scenarios must be derived from the triggering and scenario conditions identified in the course of SOTIF evaluation. These conditions are typically described only informally in natural language and are afterward converted into a machine-understandable, and sometimes formal, language. This step often involves making the vague informal description more concrete, which can be based on logical or abstract scenarios. Both options have been introduced above, and currently several industrially proven scenario description languages exist and remain under active standardization efforts [ASA24d; ASA24c]. All approaches have in common that an underlying ontology provides access to a formal and shared conceptual model of the road traffic domain (containing, e.g., a definition for what encompasses a vulnerable road user). Making the description more concrete for purposes of machine-understandability can lead to an omission of scenarios that were included

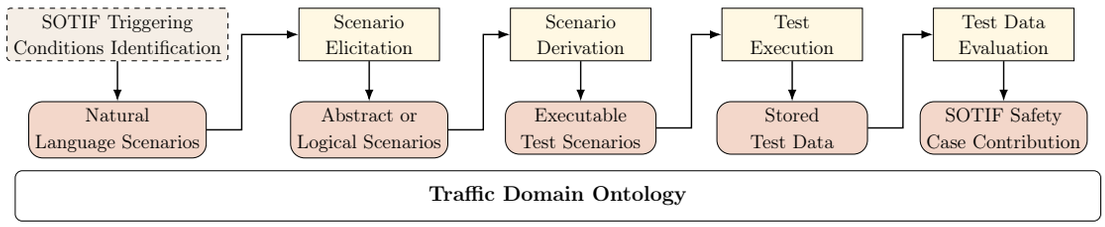

Figure 2.3: Scenario-based testing workflow in the context of SOTIF evaluation, where scenarios are defined explicitly prior to execution in the initial step of SOTIF triggering condition identification (highlighted with a dashed border). The excerpt is taken from Neurohr et al. [Neu+20, Figure 1]. Boxes with straight corners represent process steps, whereas rounded corners denote artifacts.

in the original natural language specification. In this case, it must be argued that the formal version represents the original scenario's intention adequately, as included in the challenge "SE4" mentioned by Neurohr et al. [Neu+20].

Both abstract and logical scenarios often encompass an infinite (or finite but impracticably large) set of concrete, i.e., executable, scenarios. We require techniques for instantiation, i.e., sampling a finite subset of concrete scenarios. Logical scenarios lend themselves naturally for such a task [Neu+26], and sampling can be based on established methods, including discretization of parameter spaces [AW19b; WFE20], Monte-Carlo methods [Aka+19], or optimization of target functions [PGD19]. Sampling abstract scenarios is typically harder [Neu+26], and a natural choice is using Satisfiability Modulo Theories (SMT) solvers to generate valid solutions to the specified constraint systems [Sch+19; Bor+25]. In any case, care has to be taken: Omitting chunks of the state space has to be accompanied by evidence that the selected finitely many test cases sufficiently represent the omitted scenarios. Even in case of more sophisticated probability-based approaches, Weber et al. note that usefulness depends on whether the "probability of occurrence of a scenario correlates with its relevance in terms of the safety argumentation" [WFE20]. It might even be contrarily argued that particularly the unlikely scenarios exhibit a higher criticality (due to an increased chance of them not being considered during development). Hence, a systematic exploration of the critical subspace is necessary. Neurohr et al. refer to these challenges as "TD1" and "TD4" [Neu+20, Table 1].

The sampled, concrete scenarios can now be executed using a plethora of options [Hua+16]. As with any method, there is a trade-off between validity and cost – typically, increased realism comes at the expense of higher economic burden, or, alternatively, a reduced number of test runs [Böd+18]. The cheapest option is often to solely rely on computer simulation. Validity can be increased by using mixed virtual-physical testing, where some components of the simulated system or the environment are replaced by its physical counterpart [KH24]. Highly realistic results can be achieved in real-world test

drives, including purposefully constructed proving ground setups [Lin+22] and weather chambers that allow for fine-grained control of environmental factors such as rain [Hei+19]. However, these are costly, and therefore simulation-based methods are widely adopted. In this case, scenario-based test engineers must ensure sufficient validity of the simulated models [Neu+23] (denoted by "TX1" by Neurohr et al. [Neu+20]).

The generated test data must be stored for subsequent evaluation. For simulation, data exchange formats such as ASAM Open Simulation Interface [ASA24a] can be used. For real-world data, a variety of custom data models exist, including the OMEGA format [Sch+22] or Robot Operating System (ROS) messages (the corresponding file format is called ROS bag) [Mac+22]. All of them have in common that they typically model time-stamped events on the entities under consideration, sampled at a given rate (such as 20 Hertz). Static map data can be stored using ASAM OpenDRIVE (an XML-based format) [ASA24b] or Lanelet2 (which relies on the Open Street Maps XML file format) [Pog+18]. Hence, storing the resulting test data does not pose a significant hurdle to scenario-based testing.

Finally, the collected data are evaluated, an objective this dissertation is concerned with. While scenario-based testing can also be used to assess adherence to requirements (such as always keeping a safe distance), the earlier presented use case of SOTIF quantification encompasses the computation of criticality metrics as well as factors present in the environment of the vehicle. Note that the occurrence of some of these factors might already be constrained as part of the input scenarios; however, applying simulation models can lead to violating the input constraints, which then requires re-evaluation, e.g., if the desired triggering condition is actually present in the test data. We afterward require a statistical aggregation to achieve the desired estimations of the quantities involved in Equation (2.4). While statistical estimation is out of scope, it is mentioned for understanding the broader context of the topic of the dissertation. The main challenge of statistical aggregation lies in a sound argumentation to achieve the desired confidence (referred to as "TE2" by Neurohr et al. [Neu+20]).

To conclude, we have seen that using scenario-based testing for SOTIF evaluation requires careful consideration before application. Specifically, the following fundamental considerations around scenario-based testing as presented by Neurohr et al., adapted to the context presented in Section 2.1, must be considered:

"SE4" During formalization of natural language scenarios into logical or abstract scenarios, an argument must be provided when omitting parts of the intended state space due to concretization.

"TD1" When sampling a representative set of test cases from logical or abstract scenarios, it must be argued why these offer a suitable representation of the original state space.

"TD4" Sampling methods should pay increased attention to the critical subspace of logical and abstract scenarios, and ideally explore it systematically.

"TX1" The validity of the employed simulation models must suffice for the required degree of confidence in simulation results.

"TE2" Sound statistical methods must be in place when estimating the required quantities of Equation (2.4).

2.2.2 Scenario Data Analysis for SOTIF Evaluation

We have established how data for SOTIF assessment can be *generated*. We now examine how the data can be *analyzed* regarding the presence of the events contained in Equation (2.4) – this refers to the last process step depicted in Figure 2.3 above. There are two broad categories of events that have to be evaluated: Accident and severity proxies (as a replacement for estimating harm and severity probability) as well as factors present in traffic scenarios (the hazardous events, hazardous behaviors of the vehicle, and its triggering conditions). We begin our analysis of the state-of-the-art with proxy measures in the first half of this section and continue showing how temporal logics allow for data analysis of traffic scenarios in the second half.

Criticality Metrics

The first category of data analysis in the context of SOTIF concerns accident and severity proxies. In the domain of automated driving, these are commonly referred to as *criticality metrics*. These aim to quantify criticality in a concrete scenario, i.e., recorded temporal data from, e.g., simulation or the real world. Criticality can be defined as the probability of an immediate accident of a given severity in a specific traffic constellation. Metric, in this context, does not refer to the mathematical term, but rather originates from (software) engineering. An appropriate definition can be given as "a measure of the extent [...] to which a product [...] possesses and exhibits certain [...] characteristics" [BBL76]. Transferring this to our setting, criticality metrics measure the extent to which a traffic constellation exhibits a probability of an accident of a certain severity occurring. As exemplified in Section 2.1, since critical events occur more often than actual accidents, they can reduce the required amount of collected data. The trade-off is that *valid* metrics are required in order for Equation (2.4) (which uses criticality metrics instead of harm) to be a valid approximation of risk quantification based solely on harm and severity (Equation (2.2)). We hence look at several options in this chapter and review their validity, based on what has been researched by Westhofen et al. [Wes+23]. Here, validity can be understood as the "closeness of the metrics' measurement to representing the actual accident probability and severity" [Wes+23] (often, within a specific type of scenario).

While these tools are used in various contexts throughout automated driving (and traffic safety in general) [Wes+23, Section 3], we focus here on assessing SOTIF. For completeness, however, we give a short overview of their historical development. Initially, they were introduced for studying traffic conflicts, with applications in civil engineering and vehicle safety. The idea of studying accident probability by means of traffic conflicts (similar to what has been done for Equation (2.4)) has been brought forward by General Motors in 1968 [PH68]. Under the name of *safety surrogate indicators*, such tools have quickly become popular in the 1970s, including works by Hayward [Hay72], Hydén [Hyd75], and Allen et al. [ASC78]. Several studies have provided overviews on the field's development since then [Van90; Sve98; Arc05; Mah+17].

These indicators also became popular for ADS assessment due to their general applicability for assessing the safety of traffic scenarios. Westhofen et al. have published, to the best knowledge, the first extensive literature review on the topic for the domain of automated driving in 2023 [Wes+23]. An excerpt is presented in the following.

Before diving into the literature review, a remark on the literature collection strategy is in order. Criticality metrics have been used under various names and in a multitude of fields (such as traffic accident research, psychological behavior analysis, driver assistance systems, and automated driving). Therefore, the standard method of devising a query scheme that is plugged into search engines is not deemed useful. Rather, we have assembled a diverse team of researchers and practitioners working on the topic. This combined expertise leads us to believe that the field can, in fact, be sufficiently covered.

The resulting list of criticality metrics is given in Table 2.3, along with their popularity as indicated by a search of references in OpenAlex [PPO22], conducted on 20th of March 2025. For the popularity assessment, the queries ask for

a) "advanced driver assistance system," "advanced driver assistance systems," "automated driving," "automated vehicle," "automated vehicles," "autonomous vehicle," "autonomous vehicles," "self-driving," "driverless car," "robotaxi," "robotic car," or "robo-car" in the title and abstract, and

b) the full name or the abbreviation of the metric (if applicable) in the full text.

Note that the generated ranking is not perfectly exact – for example, for Time To Brake, some works simply refer to a measured duration instead of the predictive criticality metric. However, Table 2.3 can still serve as a rough approximation for popularity.

Table 2.3: 36 criticality metrics of Westhofen et al. [Wes+23], ranked by popularity in automated driving according an OpenAlex search as of 20th of March 2025.

Metric	**No. of Results (OpenAlex)**
Time To Collision (TTC)	819
Responsibility Sensitive Safety (RSS)	153
Longitudinal Jerk (LongJ)	117
Post Encroachment Time (PET)	113
Lateral Jerk (LatJ)	110
Time Headway (THW)	109
Delta-v (Δ_v)	49
Headway (HW)	41
Time To React (TTR)	30
Time To Brake (TTB)	21
Deceleration to Safety Time (DST)	20
Required Acceleration (a_{req})	18
Brake Threat Number (BTN)	17
Time To Steer (TTS)	13
Required Lateral Acceleration ($a_{lat,req}$)	13
Proportion of Stopping Distance (PSD)	12
Time Exposed TTC (TET)	10
Crash Potential Index (CPI)	10
Time Integrated TTC (TIT)	10
Required Longitudinal Acceleration ($a_{long,req}$)	10
Steer Threat Number (STN)	8
Worst Time To Collision (WTTC)	6
Conflict Severity (CS)	6
Time To Zebra (TTZ)	5
Conflict Index (CI)	5
Time To Closest Encounter (TTCE)	4
Time To Maneuver (TTM)	4
Time To Kickdown (TTK)	3
Accepted Gap Size (AGS)	3
Distance of Closest Encounter (DCE)	3
Pedestrian Risk Index (PRI)	3
Potential Time To Collision (PTTC)	2
Predictive Encroachment Time (PrET)	2
Aggregated Crash Index (ACI)	2
Space Occupancy Index (SOI)	2
Trajectory Criticality Index (TCI)	2

We now present the ten most popular metrics, as given by Westhofen et al. [Wes+23], including their properties. For further details on the remaining metrics, consult the work of Westhofen et al. [Wes+23] or the corresponding web page[2]. The description of the metrics uses the mathematical notation given in Table 2.4. Additionally, $\|\cdot\|_2$ denotes the Euclidean norm.

Table 2.4: Mathematical notation used in the presented criticality metrics, taken from Westhofen et al. [Wes+23].

Identifier	**Meaning**
Contextual information	
$\mathcal{A}$	Set of all actors in a scene or scenario
A_i	Actor i
t	Point in time
CA	Conflict area
Entity- and time-dependent information	
$p_O(t)$	Position of object O at time t
$p_i(t)$	Position of actor i at time t
$p_{i,m}(t,t')$	Position of actor i at time t when conducting maneuver m at time t'
$d(p_1(t),p_2(t))$	Euclidean distance of $p_1(t)$ and $p_2(t)$
$d^{lat}(p_1(t),p_2(t))$	Lateral distance between $p_1(t)$ and $p_2(t)$, measured perpendicular to the heading direction of A_1
$d^{long}(p_1(t),p_2(t))$	Longitudinal distance between $p_1(t)$ and $p_2(t)$, measured along the heading direction of A_1
$v_i(t)$	Velocity of actor i at time t
$j_{i,lat}(t)$	Lateral jerk of actor i at time t
$j_{i,long}(t)$	Longitudinal jerk of actor i at time t
m_i	Mass of actor i

We distinguish between two main dimensions for classifying criticality metrics, their prediction and run time capabilities (note that more have been identified [Wes+23, Section 4]). For the first dimension, some metrics do not assess future developments and hence rely only on the current physical quantities, whereas others rely on prediction models for assessing possible future developments. The characteristics of predictive metrics highly depend on the employed prediction model. For the purpose of this dissertation, we abstract from these models and refer to Westhofen et al. for an introduction [Wes+23,

[2] https://criticality-metrics.readthedocs.io

Section 5.1]. In Table 2.4, the prediction model is incorporated by $\boldsymbol{p_i}(t)$ if t refers to a future time point. The second dimension concerns whether the metric can only be computed after all data has been fully recorded (which we then call an a-posterior metric) or whether an on-the-fly computation is possible (which we denote as an online metric).

Time To Collision TTC is an online, prediction-based, and arguably the most popular criticality metric [Hay72]. It is defined as the minimal time until an actor A_1 collides with another actor A_2 in the employed prediction model, if a future collision is predicted. Otherwise, it returns infinity. In the general case, at a given time t, it is formally defined as

$$TTC(A_1, A_2, t) = \min\left(\{\tilde{t} \geq 0 \mid d(\boldsymbol{p_1}(t+\tilde{t}), \boldsymbol{p_2}(t+\tilde{t})) = 0\} \cup \{\infty\}\right). \tag{2.5}$$

In the simplest case of a constant velocity prediction model for A_1 following A_2 with a higher speed, it reduces to

$$TTC_{const}(A_1, A_2, t) = \frac{d(\boldsymbol{p_1}(t), \boldsymbol{p_2}(t))}{\|v_1(t)\|_2 - \|v_2(t)\|_2}. \tag{2.6}$$

Several validity studies on TTC have been conducted [JLD18]. First, note that validity depends on the employed prediction model, where most works assume the constant velocity model employed in Equation (2.6). Additionally, as a rule of thumb, a more spatially distant predicted collision lowers the chance of a valid assessment. Equation (2.6) has been designed for car-following scenarios, and hence it has been found to provide an adequate representation of temporal and spatial collision characteristics along highway segments [Ozb+08].

For intersection scenarios, validity of Equation (2.5) can be reduced, as collisions have a low probability of being predicted [Wes+23, Figure 1]. Nevertheless, some works still apply Equation (2.6) to intersection settings, where it has been found that it can be used for predicting collision frequency [ES13]. TTC can also adequately represent an expert-based safety ranking of intersections with bicycle and vehicle right-turn lanes in Denmark and the Netherlands [JLD21] (however, this property does not hold for target values other than 3 and 4 seconds). However, Wang and Stamatiadis found that "TTC only identifies correctly 6 out of 18 cases [of the varying volume conditions within a simulated left turn scenario] showing that the traditional TTC indicator is far from accurate in correctly identifying the safer traffic treatment under most of the volume conditions tested" [WS14]. Additionally, Tageldin et al. have showed that TTC has not been suitable to reproduce an expert-based severity ranking of 20 motorcycle-motorcycle intersection conflicts [TSW15]. Finally, a validation has been performed on the 100-car naturalistic driving study [Din+06], which contains diverse (including urban and highway) near-miss scenarios where strong evasive maneuvers are present. Lu et al. have sorted these critical scenarios into two severity groups and selected the target value for TTC

that minimizes false positives and maximizes true positives, which was, in this case, 4.5 seconds [Lu+21]. Based on this target value, TTC has exhibited a precision of 35.9%, which can be regarded as rather low considering that the above-mentioned target value calibration method favors precision.

Responsibility Sensitive Safety Whereas TTC is popular due to its simple interpretation and straightforward implementation, the RSS takes an orthogonal direction [SSS17]. Here, a complex situational analysis is performed (incorporating factors such as road geometry) to derive safe lateral and longitudinal distances d_{min}^{lat} and d_{min}^{long}. It has been originally designed to formally guarantee safety during operation of an ADS. Formally,

$$RSS(A_1, \mathcal{A}, t) = \begin{cases} 1, \text{ if there exists } A_i \in \mathcal{A} \setminus \{A_1\} \text{ with } d^{lat}(\boldsymbol{p_1}(t), \boldsymbol{p_i}(t)) < d_{min}^{lat} \text{ or} \\ \quad d^{long}(\boldsymbol{p_1}(t), \boldsymbol{p_i}(t)) < d_{min}^{long}, \\ 0, \text{ otherwise.} \end{cases}$$

Different prediction models for intersections, highways, and unstructured roads are employed for determining d_{min}^{lat} and d_{min}^{long}, which, for the sake of brevity, are not recapitulated here. We remark that RSS relies heavily on parameters for which suitable values have to be identified before implementation. Certain cases have not been considered in the original version, such as varying road surfaces and slopes [KOW19].

A relative notion of validity for RSS has been examined for car-following and cut-in scenarios [Xu+21; Liu+21]. Here, critical events have been collected based on other criticality metrics, e.g., TTC and recorded acceleration, as well as human annotation. The studies found that RSS with parameters calibrated to such events is able to increase safety performance of a simulated ACC system, delivering some evidence on the validity of RSS relative to the employed safety performance measurements. Moreover, formal safety proofs can be done for RSS, which might serve as an additional indicator of its sensitivity [SM23].

Longitudinal and Lateral Jerk Jerk is the third derivative of position and hence the rate of change in acceleration. It can indicate maneuver abruptness, which can again serve as a proxy for the criticality of the current situation. It hence does not require a prediction model and can be computed online. The longitudinal and lateral jerk are simply defined as

$$LatJ(A_1, t) = j_{1,lat}(t), \quad LongJ(A_1, t) = j_{1,long}(t).$$

Severe collisions can be accompanied by a high jerk. LongJ has applications in assessing ACC systems. LatJ, on the other hand, indicates criticality for steering maneuvers and is therefore useful for ALKSs. As of 2018, Johnsson et al. have stated that "no previous validation studies were found for deceleration-based [including jerk] indicators" [JLD18].

We remark that the above-mentioned motorcycle-motorcycle intersection conflict study of Talgedin et al. has included, besides TTC, also jerk-based measures, which have been more consistent with the expert-based severity ranking than TTC [TSW15].

Post Encroachment Time PET is an offline, non-predictive metric and hence requires all data to be present [ASC78]. It measures the duration between an actor leaving and another actor entering a designated conflict area (CA). Here, CA can be a fixed zone (such as an intersection) or dynamically determined (such as the intersection of the paths of both actors). Under the assumption that A_1 leaves CA before A_2, it is defined as

$$PET(A_1, A_2, CA) = t_{\text{entry}}(A_2, CA) - t_{\text{exit}}(A_1, CA),$$

where t_{entry} and t_{exit} is the time of entrance and exit of CA, respectively.

Validity of PET has been examined in various works [JLD18]. For example, the above-mentioned study comparing car-bicyclist right-turn scenarios on intersections in Denmark and the Netherlands has found that, in contrast to TTC, "the PET indicator fails to agree with the ground truth at any threshold value" [JLD21]. Adding to this rather negative validity result, Lord has found in 1996 that PET did not exhibit a correlation with the estimated number of accidents between pedestrians and left-turning vehicles on four T- and four X-crossings in Canada [Lor96]. An analysis of left-turn data has revealed that the actual number of collisions of 18 intersections in the United States correlates with the amount of PET occurrences only at target values below 1.5 seconds [PHR13].

Headway and Time Headway THW is defined for car-following scenarios and measures the time until the current position of the lead vehicle will be reached [Jan05]. It hence employs a prediction model in an online setting. Assuming that A_1 follows A_2, it can be written as

$$THW(A_1, A_2, t) = \min\{\tilde{t} \geq 0 \mid \boldsymbol{p_1}(t + \tilde{t}) = \boldsymbol{p_2}(t)\}.$$

THW can be interpreted as a special case of PET, where the conflict area is simply the current position of the back of the lead vehicle. Similar to TTC, a constant velocity prediction model yields

$$THW_{const}(A_1, A_2, t) = \frac{d(\boldsymbol{p_1}(t), \boldsymbol{p_2}(t))}{\|v_1(t)\|_2}.$$

THW is related to TTC in the sense that it is equivalent to TTC under zero speed of the lead vehicle, i.e., $\|v_2(t)\| = 0$. It thus represents a partial worst-case TTC (note that constant velocity of the following vehicle is still a non-worst-case assumption). This highlights its potential for higher sensitivity compared to TTC. Another indication of its validity is given by its use in legislation, where THW can be used for imposing fines [Vog03; Lüt+25].

HW is a related metric used in several traffic regulations and is simply the distance to the lead vehicle [Jan05]:

$$HW(A_1, A_2, t) = d(\boldsymbol{p_1}(t), \boldsymbol{p_2}(t)).$$

As it disregards dynamic information, its validity compared to THW is reduced.

Delta-v While the above metrics approximated mainly collision probability, Δ_v on the other hand is a severity-focused criticality metric. It measures the change in speed that happens during a collision [Car79; She11]. In the classical sense, it is an a-posterior metric (using accident data) without a prediction model. However, an online version can be devised by means of a post-collision velocity prediction model (in the simplest case, it always predicts a velocity of zero). Formally, it is given as

$$\Delta v(A_1) = \|v_1(t_{aftercol}) - v_1(t_{beforecol})\|_2,$$

where $t_{aftercol}$ and $t_{beforecol}$ are the times directly after and before collision, respectively. The core idea is to relate the dissipated energy to the severity of obtained injuries. In fact, it was found that the probability P of fatality for actor A_1 in two-vehicle collisions can be expressed using Δ_v as [Jok93]

$$P(A_1) \approx \left(\frac{\Delta v}{31.74\text{m/s}}\right)^4.$$

While other severity proxies exist, this model has been found surprisingly valid for two-vehicle collisions, given its simple computation [She11]. An extended version additionally takes masses into account [She11].

Time To Brake and Time To React TTB can be seen as a special case of the more general TTM metric [HSK06; TDB11]. TTM is defined as the latest future time point up to TTC for which a given maneuver m conducted by A_1 would still avoid a collision with A_2. If collision is unavoidable, it returns $-\infty$. Formally,

$$TTM(A_1, A_2, t, m) = \max\left(\{\tilde{t} \in [0, TTC(A_1, A_2, t)] \mid \right.$$
$$\left. d(\boldsymbol{p_{1,m}}(t+s, t+\tilde{t}), \boldsymbol{p_2}(t+s)) > 0 \; \forall\, s \geq \tilde{t}\} \cup \{-\infty\}\right).$$

TTB is then defined as TTM under the maneuver "brake," i.e., the latest time an emergency braking maneuver can prevent a collision. To the best knowledge, no studies concerning the validation of TTB exist. However, comparing TTB to other TTM-based metrics, it has to be noted that braking is often a natural human emergency reaction [Ada94], and hence TTB can offer valuable insights into the criticality of human-driven vehicles. For ADSs however, it has to be remarked that non-braking evasive maneuvers can successfully

avoid critical situations even if there is no time left to brake, i.e., TTB is zero [Ada94]. Maneuvers besides braking should thus be considered for increased validity when assessing ADSs.

TTR [HSK06] extends TTM by aggregation over a set of possible maneuvers M:

$$TTR(A_1, A_2, t, M) = \max_{m \in M} TTM(A_1, A_2, t, m)\,.$$

The original work has examined example scenarios as initial evidence for the validity of TTR and claims it to be "an adequate metric to assess the criticality of traffic situations since it directly relates to the driver's possible actions" [HSK06]. However, there appears to be no work that assesses its validity on diverse data, and it generally depends on the employed maneuvers and prediction models [Wes+23].

Logics-Based Data Querying

Besides computing criticality metrics as proxies for harm and severity used in Equation (2.4), we also require an estimate of the identified SOTIF triggering conditions, hazardous behaviors, and hazardous events. In the given Example 2.1.2, these factors include "ice plates falling from truck in front," an "unsafe evasive maneuver," as well as "swerving with ice on road with multiple lanes." In contrast to the above-introduced criticality metrics, these factors relate to more abstract happenings in traffic, such as weather conditions, road network configurations, and spatio-temporal evolutions of actors within these networks. The physics-focused view of criticality metrics offers a solid foundation when referring to the quantities mentioned in Table 2.4 (such as positions and velocities), but behaves cumbersome when trying to formalize more abstract traffic relations.

In what follows, we show that for the task of identifying such abstract conditions in recorded traffic data, logics have proven to be a valuable tool. Based on the identified related work, Table 2.5 lists over 20 publications from five broad categories of logic-based frameworks: formal languages (such as context-free grammars), temporal logics (including LTL_f), logic programming, e.g., Horn clauses, DLs, and semiformal approaches. Generally, the earliest listed work in the area has been conducted on logic programming, DLs, and formal languages. The first identified article on applying temporal logics in the subject area has been published only in 2017.

Table 2.5: Related work on logic-based data querying for automated driving, including whether it allows for ontological background knowledge (Ont. Know.) and incorporates the Open World Assumption (OWA).

Reference	Temporal Formalization Means	Nontemporal Formalization Means	Ont. Know.	OWA
		Semiformal		
[Gel+20]	Sequences	Computable functions	✗	✗
[Gel+22]	Sets of sequences with event triggers	OpenSCENARIO XML conditions	✓	✗
		Formal Languages		
[Hül+10]	Probabilistic FA	Fuzzy Control Language	✓	✗
[Luc+16]	Context-free grammars	Terminal symbols	✗	✗
[Els+20]	Regular expressions	Propositions	✗	✗
		Temporal Logics		
[Riz+17]	LTL (finite semantics)	HOL	✗	✗
[Aré19]	STL (robust semantics)	Nonlinear real arithmetic	✗	✗
[Mai+20], [MMA22]	Discrete-time MTL (finite intervals)	FOL	✗	✗
[GA21]	STL (past only, robust semantics)	FOL	✗	✗
[Est22]	LTLf	Propositions	✗	✗
[Mat+22]	LTL with past	Modal metric spaces	✗	✗
[Sch+24b]	Extended MFOTL	Computable functions	✗	✗
[Tol+24]	LTLf	Propositions (derived from scene graphs)	✗	✗
[Ste+25]	First-order multi-sorted real-time logic	First-order linear real arithmetic	✗	✗
		Logic Programming		
[NR03]	(Fuzzy) Horn clauses	(Fuzzy) Horn clauses	✓	✗
[MN12]	SWRL	SWRL	✓	✓
[Zha+16]	✗	SWRL	✓	✓
[KD20]	ASP	ASP	✓	✗
		Description Logics		
[Hum09]	✗	$\mathcal{SHIQ}$	✓	✓
[HZW11]	✗	$\mathcal{SHIQ}^{(\mathcal{D})}$	✓	✓
[AFI14]	$\mathcal{SROIQ}^{(\mathcal{D})}$	$\mathcal{SROIQ}^{(\mathcal{D})}$ & SWRL	✓	✓
[Bue+17]	✗	$\mathcal{SROIQ}^{(\mathcal{D})}$ & SWRL	✓	✓

Some works approach the topic in an informal or semiformal manner, where not all aspects of the data querying language are formalized. This dissertation focuses on fully formal frameworks for reasons discussed in Section 2.3. We thus consider such approaches out of scope, but shortly mention the work of Gelder et al. [Gel+20; Gel+22] as two prominent examples. In their first work, they have queried data for the occurrence of tag sequences, where the data are annotated by tags which are defined via arbitrary computable functions. This idea has been generalized to sets of sequences, where each sequence can be triggered by events implemented as OpenSCENARIO XML conditions [Gel+22]. However, in both works, no formalism underlying the semantics of sequences has been given, but the second line of work allows for specifying ontological background knowledge within the provided object-oriented domain model.

A natural choice for enabling a formal semantics for the temporal part of the query language is the use of formal languages. The main idea is to represent the finite, discrete data traces as words over a suitable alphabet, which enables using the standard formalisms for encoding formal languages. These include probabilistic FAs [Hül+10], context-free grammars [Luc+16], and regular expressions [Els+20], formalisms with varying degrees of expressiveness. Besides expressiveness, practical considerations are often taken into account – such as efficiency and software availability. Regular expressions (or, equivalently, FAs), as used by Elspas et al. [Els+20], offer low computational complexity while users might already be familiar with the framework. Luccetti et al. have leveraged the more general formalism of context-free grammars [Luc+16], however, all the provided example grammars appear to be right-linear and hence specifiable as regular expressions, too. The mentioned approaches have in common that basic events are extracted from the data first, and the temporal formalism is evaluated afterward. For example, Hülnhagen et al. have relied on the so-called Fuzzy Control Language – a rule-based logic system – to derive propositions from the given data signals [Hül+10]. Such rules can also be used to specify background knowledge; however, they are interpreted under a Closed World Assumption (CWA).

A related line of work examines the use of temporal logics such as LTL, Signal Temporal Logic (STL), and Metric Temporal Logic (MTL) in place of directly employing formal grammars and automata. Similarly to the frameworks above, these logics are typically interpreted over finite words. The first approach was proposed by Rizaldi et al. in 2017 [Riz+17], who have used Higher Order Logic (HOL) (which captures their employed variant of LTL) and the Isabelle theorem prover to derive software code representing the specified traffic scenarios. Subsequently, many works investigated extensions or derivatives of LTL. For example, Maierhofer et al. have relied on discrete-time MTL formulae, where predicates are defined by First Order Logic (FOL) expressions [Mai+20; MMA22]. The first identified work employing a spatio-temporal logic for automated driving data analysis have been published in 2022 [Mat+22], where, instead of the usual predicates, spatial

constraints can be expressed in LTL subformulae. However, both LTL and discrete-time MTL are typically interpreted over infinite traces. Since traffic scenarios are of finite duration, Rizaldi et al. have modified LTL semantics, Maierhofer et al. have restricted MTL syntax to finite intervals, and Pedro et al. have incorporated past operators into LTL.

Alternatively, one can directly rely on a finite-trace temporal logic, including STL, which syntactically extends LTL by interval constraints and inequalities over real-valued signals, and $\mathrm{LTL_f}$, which corresponds with star-free regular expressions [GV13]. For example, Arechiga have employed STL, and, to specify nontemporal constraints, added nonlinear real arithmetic to the default setting of propositions and inequalities over signals [Aré19]. Similarly, Gressenbuch and Althoff have relied on FOL-like expressions to define predicates for STL formulae in order to represent traffic rules [GA21]. An early use of $\mathrm{LTL_f}$ for the examined application domain is identifiable in 2022, where Esterle has used it for specifying traffic rules [Est22, Chapter 4]. Whereas Esterle has given only a natural language definition for each predicate, Toledo et al. have proposed to derive the presence of $\mathrm{LTL_f}$ predicates from scene graphs [Tol+24].

There furthermore exist custom temporal logics specifically designed for the demands of traffic scenario specification and data querying. These include the Counting Metric First-Order Temporal Binding Logic (CMFTBL) as defined by Schallau et al. [Sch+24b], an extension of Metric First-Order Temporal Logic (MFOTL) devised for scenario classification with the goal of measuring test coverage. Another example concerns Traffic Sequence Charts (TSCs), which are semantically defined as a first-order multi-sorted real-time logic [Ste+25]. As a spatio-temporal logic, TSCs allow for first-order linear real arithmetic constraints to define spatial invariants, however, as defined by Stemmer et al., do not allow the user to specify ontological background knowledge. In general, none of the above introduced approaches to using temporal logics offer means to incorporate such knowledge, and, additionally, a closed world interpretation is made in all cases.

An orthogonal research direction concerns the use of knowledge representation and reasoning systems, specifically, logic programming and DLs. Since knowledge representation is a primary goal in these formalisms, all the subsequently introduced works in this area allow incorporating an ontology, and many make the OWA. Rule systems, such as Horn clauses and the Semantic Web Rule Language (SWRL), are common tools used in logic programming. Already in 2003, Nigro and Rombaut have proposed the IDRES system to recognize traffic situations (such as overtaking maneuvers) [NR03]. It uses fuzzy Horn clauses to derive from sensor data a set of states, which is then passed into a second set of rules representing temporal connectives. Using the same formal framework for temporal and nontemporal specification is also employed by Morignot and Nashashibi [MN12]; here, SWRL is used. SWRL works on top of DLs and thus assumes an open world. This has been leveraged by the authors to specify a background taxonomy. Similarly, Zhao et al.

have used SWRL for real-time situation assessment, but do not incorporate special means to handle temporal specifications [Zha+16]. In both works, the reasoner Pellet is relied upon, which is also the foundation of the software tool presented in this dissertation. A different route is taken by Karimi and Duggirala, who have proposed the use of Answer Set Programming (ASP) to specify and evaluate traffic rules [KD20]. Since ASP does not include temporal aspects by default, Karimi and Duggirala have manually encoded temporal axioms in ASP, which corresponds to what has been done by Morignot and Nashashibi for SWRL as well.

The final category in Table 2.5 concerns DLs as a formalization tool. This dissertation also argues for DLs to specify nontemporal knowledge. But, in contrast, the subsequently presented works either provide no temporal reasoning mechanism or employ DLs also for this task. One of the earliest identifiable works on using DLs for situation recognition for automated driving has been published by Hummel in 2009, who has focused on spatial and road network reasoning [Hum09]. The ontology has been successfully used for classification and consistency checking of data from the AnnieWAY vehicle, a finalist in the 2007 DARPA Urban Challenge. Hülsen et al. have used a leaner approach (without extensive spatial reasoning) to query driving data for various relations between traffic participants and road infrastructure [HZW11]. In their experiments, they have achieved reasoning frequencies below one Hertz, which did not meet their desired real-time frequency of two Hertz. Both Hummel and Hülsen et al. have not incorporated temporal reasoning. In contrast, Armand et al. have shown how $\mathcal{SROIQ}^{(\mathcal{D})}$ axioms and SWRL rules can be used for situation recognition in real-time while also including temporal knowledge [AFI14]. Buechel et al. have showcased how such a DL-based formalization approach can be leveraged for encoding rich legal background knowledge on traffic regulations and evaluate these rules on data [Bue+17]. By this, they have provided a strong argument in favor of incorporating ontologies into data querying. Note that all the referenced approaches have used highly expressive DLs, a setting that is also assumed in the dissertation at hand. While Hummel and Hülsen have taken the then popular RacerPro system, Armand et al. and Büchel et al. have transitioned to the DL reasoner Pellet. DLs apparently provide expressive means to handle background knowledge in the traffic domain, but it appears that the identified related work misses practicable mechanisms to specify temporal aspects.

In conclusion, the above-sketched research proposes a plethora of options for logic-based situation recognition and data querying in automated driving applications. Still, it currently appears that no option combines the three facets argued for in the following Section 2.3: a built-in temporal logic mechanism, means to specify ontological background knowledge, and employing an OWA.

2.3 The Gap – Advancing Logics-Based Scenario Data Analysis

In Section 2.1, we have seen that central regulations reference ISO 21448, which has the assessment of SOTIF as its main concern. A possible way of estimating SOTIF involves generating data using scenario-based testing and subsequently applying criticality metrics and temporal logics for data analysis. A review of the state-of-the-art showed that a plethora of metrics exists readily available for the practitioner. While novel criticality metrics are still being developed, the existence of software tools and frameworks provide evidence for maturity of the field [LA23; Lüt+25]. Logic-based analysis of traffic data, on the other hand, is comparatively novel. Taking into account the state-of-the-art on logic-based data analysis, we now devise requirements on what is missing for a thorough logics-based scenario data analysis in the context of SOTIF, on the grounds of the research initiated by Westhofen et al. [Wes23b; Ste24; Wes+24a]. We do so by systematically analyzing the task at hand. In all brevity, we are confronted with relational, temporal, and possibly incomplete data that has to be interpreted under extensive background knowledge for correct analysis, and desire a logic-based approach for traceable and consistent specifications.

Data Section 2.2.1 already gave indications towards the shape of generated data: It consists of a finite sequence of time-stamped snapshots of the system and its environment. Each snapshot contains a possibly different set of entities (or individuals) present in the currently observable environment. Concrete physical data can be attached to these individuals, such as geometries and velocity vectors. Additionally, individuals are often annotated with abstract information, including their type, state, condition, and material. Information can not only be attached to individuals but also exists between individuals, e.g., whether an individual is occluded for another observing entity, which lane a traffic sign or light legally applies to, or how roads and lanes are connected to each other. For an industrially relevant case study containing such information, consider the OMEGA data format [Sch+22]. It has been successfully used to analyze real-world drone data [Wes+22a] but can also be employed to represent simulation results [SGE24].

In case of real-world data recordings, we are confronted with the additional issue of imperfect sensor hardware. Even if sensors had no faults, the physical reality (such as occlusions or inferences in the electromagnetic spectrum) implies that the data might not catch all information present in the real world. Hence, any perception is only partial w.r.t. the actual state of the world. Despite partial information, we still aspire correct data analysis without incorrect conclusions. For this, the data must be interpreted under the OWA: If something is not explicitly stated in the data, it cannot be inferred to not exist. On the contrary, if we are sure that something cannot exist, this fact must be modeled in

the data or at least be derivable from background assumptions.

Finally, we are interested in querying the data, i.e., fetching a list of individuals satisfying the constraints, as opposed to merely checking whether the Boolean constraint holds. It is often relevant to assess *which* individuals in the data contribute to a critical situation in order to establish a causal link between identified events. Consider the example hazard model of Figure 2.2, where a connection was drawn between ice plates being on a truck (a triggering condition) and an inappropriate distance (a hazardous event). Both conditions are modeled as separate queries, however, we must check if they refer to the same individuals for validly determining risk.

Background Knowledge The above explanation of the OWA already hints towards another required component: relying on background knowledge to correctly interpret the data. In-depth background knowledge must be included since situations may otherwise not be correctly recognized in the data and a test evaluation produces false results. For this, *ontologies* are a popular tool in automated driving applications, see, e.g., ASAM OpenXOntology [ASA22] for an international standardization project as well as Westhofen et al. [Wes+22a] and Zipfl et al. [ZKZ23] for non-systematic reviews. In computer science, an ontology refers to a formalized conceptualization of a domain of discourse [LÖ18]. These can range from simple subsumption axioms up to highly involved relations extracted from legal texts. As written above, we do not assume that the data are complete in the sense that we can comprehensively observe all facts about all individuals. Instead, we assume to have rich knowledge about the relations used in the situation descriptions. An example for such knowledge is to assert that humans mounted on vehicles (such as bicycles) can never be pedestrians, and pedestrians have right of way on pedestrian crossings. In this way, if a bicyclist appears on a pedestrian crossing, it can be inferred that right of way does not have to be granted.

Another demand concerns iterative specifications: For example, assume that an engineer specified a sufficient condition for parking vehicles based on dynamical objects, albeit there may be parking vehicles that are no dynamical objects. However, at the time of specification we can safely ignore this matter, as, per the OWA, an individual cannot be inferred to *not* be a parking vehicle just because it *cannot* be inferred that it is one. Axioms such as the inclusion of broken vehicles might be added in later development phases, e.g., based on discussions with legal experts. These additional axioms should not break with inferences over data obtained from earlier versions of the specification, i.e., we require monotonicity. Otherwise, upstream data analysis results cannot be re-used downstream in the development process. A formal definition of monotonicity in our context is given in Section 3.2.1.

Logics The final requirement concerns a logic-based specification. While not strictly necessary, using logics has several advantages over informal specification languages:

- the formal semantics allows for an unambiguous specification of background knowledge across stakeholders from multiple domains and serves as a unifying basis for all involved tooling,
- created formulae can be automatically checked, e.g., for consistency, allowing to detect modeling errors more easily, and
- it can be formally proven that the returned data analysis results correctly reflect the input specification.

Many of the logic-based approaches presented above specifically incorporated aspects of temporal logics. For these, additional requirements can be gathered: As SOTIF evaluation allows us to simply record temporal data and analyze them retrospectively, we are not in a run-time context. This means that neither real-time nor streaming capabilities are necessary and, in general, operating under restricted computational resources is not a primary concern. Moreover, such an offline analysis setting does not demand the use of past-time operators. This is in contrast to using data querying for implementing run-time situation awareness capabilities into the system, where mixed past- and future-time operators might be of value. On the other hand, duration constraints are often required during specification, e.g., to distinguish actions of certain lengths (such as giving upper bounds on the legal duration of an overtaking maneuver). Therefore, we require metric temporal operators.

Requirements To conclude, we desire a logic-based mechanism that incorporates complex background knowledge to query for the presence of specified situations in temporal, relational data under the OWA. By this, we arrive at the following required (non-)features for a query language that can be used for identifying SOTIF-relevant factors in recorded traffic scenarios:

1. The data model must allow representing temporal data of finite duration and consists of finitely many time-stamped snapshots. Each snapshot contains a finite amount of information on individuals, properties, and their relations.
2. Interpretation of this data must consider the data to be incomplete, due to inherent properties of the employed sensors.
3. Data must be queried – a Boolean result is often not enough.
4. We require expressiveness in the background knowledge modeling language, as we benefit from strong inferences on the domain model and have loose computational constraints.
5. The model used for background knowledge must permit adding to existing specifications without invalidating already inferred facts (monotonicity).

6. A logic-based approach allows benefiting from the many advantages of formal rigor, including semantic preciseness and consistency analyses.

7. We require only future-time temporal operators, and, as we are not in a streaming setting, past-time operators can be ignored. On the other hand, metric operators are necessary for specifying constraints on durations.

8. The queries will be computed offline, and hence run-time or streaming capabilities are not required.

These requirements lay the foundation of the query language presented in Chapter 3.

3 A Temporal Query Language

Contents

This chapter defines a novel solution for recognizing relevant situations in temporal data, derived from the applied perspective delineated above. The framework is built on existing work: specifically, research on expressive DL ontologies and CQs, which are both introduced in Section 3.2 after some preliminaries in Section 3.1. Despite these sections

not presenting novel developments but rather existing formalisms and works, they form the necessary foundation for the remainder of this chapter. As an advancement over existing literature, Section 3.2 adds a review of CQ answering from a theoretical and practical perspective, explaining in a condensed manner how tractability was achieved. Moreover, an extended overview of the state-of-the-art in temporal ontology-based querying in Section 3.3 is provided as a minor contribution.

The main contribution of this chapter is the temporal query language presented in Section 3.4, based on the work of Westhofen et al. published in 2024 [Wes+24a]. A related setting has been examined by Baader et al. [BBL13], however, their work is concerned with theoretical aspects, such as decidability. The dissertation focuses on formulating a practical query language over expressive DLs for which efficient algorithms can be developed. To the best of knowledge, such a language has not been proposed in the literature.

3.1 Preliminaries

Before we dive into the technical part of this dissertation, we discuss some preliminaries the reader should be familiar with, specifically pertaining to automata theory and computational complexity.

3.1.1 Notation

Throughout this dissertation, we use several notation conventions. For natural numbers m and n, i.e., $m, n \in \mathbb{N} = \{0, 1, \dots\}$, a *finite sequence* $(a_i)_{i \in \{m,\dots,n\}} = (a_m, \dots, a_n)$ is an ordered enumeration of $n - m + 1$ elements. If $n < m$, we obtain the empty sequence. The *Cartesian product* of two sets A and B is denoted by $A \times B = \{(a, b) \mid a \in A,\, b \in B\}$, and its n-fold application on A is given by $A^n = A \times \cdots \times A$. The *disjoint union* of two sets is defined as $A \uplus B = \{(a, 0) \mid a \in A\} \cup \{(b, 1) \mid b \in B\}$. For a function $\pi : A \to B$, $\mathrm{Im}(\pi)$ refers to the *image* $\{\pi(a) \mid a \in A\}$ of π. A *labeled graph* G is a tuple $G = (V, E, L)$ where V is a set of vertices, L a set of labels, and the edges $E \subseteq V \times L \times V$. Note that we only consider directed, labeled graphs. If two formulae φ and ψ are semantically (but not necessarily syntactically) *equivalent*, we sometimes write $\varphi \equiv \psi$.

For clarity, we rely on some specific notation for KBs and queries. Sets related to both formalisms are written in sans-serif font, e.g., the set of answer variables of a query Φ, $\mathsf{Var}(\Phi)$, or the set of individuals of a KB $\mathcal{K}$, $\mathsf{Ind}(\mathcal{K})$. We denote possible answers to a query by using vector notation, e.g., $\vec{a} \in \mathsf{Ind}(\mathcal{K})^{|\mathsf{Var}(\Phi)|}$. Analogously, query variables are – when ordered as a tuple – written as $\vec{x}$. Individual, concept, and role names of a KB are rendered in monospaced font, e.g., `Joe`, `Driver`, or `drives`. In general, non-temporal queries are represented by lowercase letters, e.g., φ and ψ. We use uppercase for temporal queries, e.g., Φ and Ψ.

Finally, names of software tools are written in capital letters, e.g., TOPLLET, and classes, modules, and terminal commands of software in monospaced font, e.g., `topllet help`.

3.1.2 Formal Languages and Automata

An *alphabet* Σ is a non-empty, finite set of letters. We can build a finite word $w = a_1 a_2 \dots a_n$ of length $n \in \mathbb{N}$ by concatenation of letters. The length of a word is denoted by $|w| \in \mathbb{N}$, where for the empty word $\epsilon \notin \Sigma$ it holds that $|\epsilon| = 0$. We express the number of occurrences of a letter $a \in \Sigma$ in a word w as $|w|_a$. Note that for two words u, v, it holds that $|u|_a + |v|_a = |uv|_a$. Similarly to letters, two words u and v can be concatenated as uv.

A *(formal) language* L over Σ is a subset of Σ^*, which denotes the set of all finite words one can construct from Σ. Here, the *Kleene operator* $\cdot^*$ is defined for a language L as $L^* = \bigcup_{i \in \mathbb{N}} L^i$ where $L^0 = \{\epsilon\}$ contains only the empty word ϵ, $L^{i+1} = L \cdot L^i$, and $L_1 \cdot L_2 = \{uv \mid u \in L_1, v \in L_2\}$. Similarly, for a word $w \in \Sigma^*$ and $n \in \mathbb{N}$, we define $w^n = ww^{n-1}$ if $n > 0$ and $w^0 = \epsilon$.

Sometimes, we will use *regular expressions* to concisely represent a language over an alphabet Σ. Regular expressions are built from the constants of Σ for which the language of the regular expression is simply the letter itself: $L(a) = \{a\}$ for $a \in \Sigma$. Two regular expressions r and s can be concatenated as rs, where $L(rs) = \{uv \mid u \in L(r), v \in L(s)\}$, and alternated as $r \vee s$, where $L(r \vee s) = L(r) \cup L(s)$. Similar to above, we define the Kleene operator on regular expressions as r^* to denote the language $L(r)^*$.

A *(deterministic) Finite Automaton (FA)* is a tuple $A = (Q, \Sigma, q_0, \delta, F)$ where Σ an alphabet, Q is a finite set of states, $\delta : Q \times \Sigma \to Q$ is a transition function, $q_0 \in Q$ the initial state, and $F \subseteq Q$ the final states. The language accepted by a deterministic FA A is defined as $L(A) = \{\tau \in \Sigma^* \mid \delta^*(q_0, \tau) \in F\}$, where $\delta^* : Q \times \Sigma^* \to Q$ with $\delta^*(q, \epsilon) = q$ and $\delta^*(q, wa) = \delta(\delta^*(q, w), a)$ for $w \in \Sigma^*$ and $a \in \Sigma$. A path of length n is a sequence $q_0 \overset{a_0}{\to} q_1 \dots q_{n-1} \overset{a_{n-1}}{\to} q_n$ where for each $q_i, q_{i+1} \in Q$, $0 \leq i < n$, and $a_i \in \Sigma$ it holds that $\delta(q_i, a_i) = q_{i+1}$. We write $q_0 \to^n q_n$ if there exists a path of length n from q_0 to q_n. A is minimal if there is no other deterministic FA $A' = (Q', \Sigma, q_0, \delta', F')$ with $|Q'| < |Q|$. The minimal deterministic FA is unique and can be computed in polynomial time [HMU07].

A *nondeterministic FA* uses a transition relation $\delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$ instead of a function, where we also allow the empty word ϵ in transitions. A path of length n through a nondeterministic FA is a sequence $q_0 \overset{a_0}{\to} q_1 \dots q_{n-1} \overset{a_{n-1}}{\to} q_n$ where $(q_i, a_i, q_{i+1}) \in \delta$ for any $q_i, q_{i+1} \in Q$, $0 \leq i < n$, and $a_i \in \Sigma \cup \{\epsilon\}$. Analogously to deterministic FA, $L(A) = \{\tau \in \Sigma^* \mid \delta^*(q_0, \tau) \cap F \neq \emptyset\}$, where $\delta^* : Q \times \Sigma^* \to 2^Q$ with $\delta^*(q, wa) = \bigcup_{p \in \delta^*(q,w), (p,a,p') \in \delta} \delta^*(p', \epsilon)$. To account for ϵ-transitions, we define the ϵ-closure $\delta^*(q, \epsilon) = \{p \in Q \mid \text{there is a path } q = q_0 \overset{\epsilon}{\to} \dots \overset{\epsilon}{\to} q_n = p, n \geq 0\}$. It is known that for any nondeterministic FA A, there is a deterministic FA A' s.t. $L(A) = L(A')$ [HMU07].

3.1.3 An Introduction to Computational Complexity

Computational complexity is concerned with the time and space resources of algorithms that solve a particular problem. There are two general views on computational complexity: First, examining algorithms themselves, and second, the analysis of the underlying problem, i.e., the inherent complexity of the problem regardless of the employed algorithm. We will adopt the first view in Chapter 4, where we analyze the performance of the presented algorithms, and sometimes use the *Big O notation* to denote asymptotic run time. In all brevity, we define, for a function $f : \mathbb{N} \to \mathbb{R}_{\geq 0}$, $O(f(n)) = \{g(n) : \mathbb{N} \to \mathbb{R}_{\geq 0} \mid \exists c > 0, n_0 \geq 0.\ \forall n \geq n_0.\ g(n) \leq c \cdot f(n)\}$, i.e., the set of all functions whose infinite behavior is bounded by $f(n)$ when ignoring constant factors. To avoid confusion with ontologies – which are denoted by $\mathcal{O}$ – we stick with the plain symbol O for asymptotic run time in this dissertation.

For Chapter 3, we focus on the more abstract, second view on computational complexity. Here, we often assess *complexity classes*, where problems with comparable resource demands are bundled. We typically examine *decision problems* for this, which require computing a binary decision ("yes" or "no") given some input. These problems can be characterized in terms of lower and upper bounds on their complexity class: the lower bound states that one cannot devise a better algorithm with time or space constraints described by the class; the upper bound can simply be given by an algorithm that runs within the specified resource constraints. In the latter case, we assign the given decision problem into the complexity class. An example of a complexity class is P, which contains all problems that can be deterministically solved in polynomial time, or, more formally, for which a deterministic algorithm with a run time in $O(p(n))$ for an input of size n and a polynomial $p(n)$ exists. Several other complexity classes appear throughout this dissertation. For an input size n and polynomial $p(n)$, we define

- NP as the set of all decision problems that can be solved by a nondeterministic algorithm using a run time in $O(p(n))$,
- ExpTime as the set of all decision problems that can be solved by a deterministic algorithm using a run time in $O(2^{p(n)})$,
- 2ExpTime as the set of all decision problems that can be solved by a deterministic algorithm using a run time in $O(2^{(2^{p(n)})})$,
- PSpace as the set of all decision problems that can be solved by a deterministic algorithm using memory of size $O(p(n))$, and
- NSpace($O(n)$) as the set of all decision problems that can be solved by a nondeterministic algorithm using memory of size $O(n)$.

The set of *hard problems* of a given complexity class is especially relevant – they are defined as those problems for which a polynomial-time reduction from any problem in the complexity class exists. We say that a decision problem is *complete* w.r.t. a complexity class if it is both hard and contained in this class. There exist natural complements of the complexity classes, e.g., co-NP, which is just the set of all decision problems whose complement is in the specific complexity class, e.g., NP. A special class, which will later be used, is D^p, whose problems can be represented as the intersection of an NP- and co-NP-problem. Formally, $D^p = \{L_1 \cap L_2 \mid L_1 \in \text{NP}, L_2 \in \text{coNP}\}$.

In the field of querying data, an additional distinction is made between *combined complexity* – which has been described above – and *data complexity* [Var82]. Data complexity assumes that only the data component is part of the input size n and sets all other components, such as they query and ontology, as constant. For example, answering CQs in the expressive DL $\mathcal{SHIQ}$ is co-NP-complete w.r.t. data complexity [Gli+08]. This allows a more realistic description of the behavior in cases where very large data are given. However, this assumption is hard to make in the examined practical setting of this dissertation, as the size of in-depth domain ontologies might outweigh a couple of seconds of temporal data sampled with ten Hertz. This dissertation is therefore mainly concerned with combined complexity.

3.2 Querying in Description Logics

The temporal query language introduced in this dissertation is based on DLs, a knowledge representation formalism with roots in the 1980s, which originated from the desire to give formal semantics to frames and semantic networks [BHS08]. We start by presenting the standard syntactical and semantical description of what has emerged as the expressive DLs $\mathcal{ALC}$ and $\mathcal{SROIQ}^{(\mathcal{D})}$, as well as its relevant decision problems. The latter DL has been successfully put into practice by the OWL2 standard, for which we also give a brief introduction. CQ answering over DLs – an established means for OBDA – is introduced subsequently both from a theoretical and practical perspective. This section hence merely collects and presents existing work, which will form the foundation for our custom temporal query language presented in Section 3.4.

3.2.1 Syntax and Semantics of Description Logics

Fundamentals of Description Logics

We start by introducing the syntax and semantics of DLs. The following definitions are all based on a standard reference from the DL literature [Baa+07, Chapter 2.2]. In DLs, we can describe the relationship of roles and concepts in an ontology $\mathcal{O}$ and assert individuals to these concepts and roles in a database $\mathcal{D}$. A prototypical DL fragment is

$\mathcal{ALC}$, which stands for *Attributive Language with Complements*. This fragment will serve as a prototype for the majority of our formal introduction. For the remainder, we fix countably infinite supplies $\mathsf{N_C}, \mathsf{N_R}, \mathsf{N_I}$ of concept, role, and individual names, respectively. In what follows, we write elements of these supplies in typewriter font, e.g., $\mathtt{C} \in \mathsf{N_C}$. Moreover, sets and functions mapping to sets are denoted in sans-serif, e.g., $\mathsf{N_C}$.

We begin with $\mathcal{ALC}$ concepts, which form the backbone of formalizing abstract domain knowledge over the given concept, role, and individual names. Later, we will introduce more expressive DL fragments.

Definition 3.2.1 ($\mathcal{ALC}$-Concept)

An $\mathcal{ALC}$-concept C adheres to the grammar

$$C ::= \mathtt{A} \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall \mathtt{r}.C \mid \exists \mathtt{r}.C$$

where $\mathtt{A} \in \mathsf{N_C}$ and $\mathtt{r} \in \mathsf{N_R}$.

We can thus compose new concepts using negation, intersection, and union. For a role $\mathtt{r}$, we allow for universal (enforcing a concept to only have $\mathtt{r}$-successors in $\mathtt{C}$) and existential quantification (enforcing a concept to have an $\mathtt{r}$-successor in $\mathtt{C}$). Syntactically, we define symbols for the special concepts of all things $\top := \mathtt{A} \sqcup \neg \mathtt{A}$ for some $\mathtt{A} \in \mathsf{N_C}$ and nothing $\bot := \neg\top$.

Example 3.2.1

A very basic $\mathcal{ALC}$-concept is `Pedestrian`, which can, for example, be used in compound concepts such as $\mathtt{Driver} \sqcup \mathtt{Pedestrian}$, referring to any individual that is a driver or pedestrian.

The central mechanism for modeling knowledge in $\mathcal{ALC}$ are concept inclusions, i.e., the subsumption relation between concepts.

Definition 3.2.2 (Concept Inclusion)

A concept inclusion has the form $C \sqsubseteq D$ for $\mathcal{ALC}$-concepts C and D.

As a shorthand notation, we write $\mathtt{C} \equiv \mathtt{D}$ (concept equivalence) for $\mathtt{C} \sqsubseteq \mathtt{D}$ and $\mathtt{D} \sqsubseteq \mathtt{C}$.

Example 3.2.2

The knowledge that humans are exactly pedestrians or drivers can be expressed as the concept equivalence $\mathtt{Human} \equiv \mathtt{Driver} \sqcup \mathtt{Pedestrian}$. Other example knowledge is crowd members being humans next to other humans, and crowd members never being drivers: $\mathtt{CrowdMember} \equiv \mathtt{Human} \sqcap \exists \mathtt{next\_to}.\mathtt{Human}$ and $\mathtt{CrowdMember} \sqsubseteq \neg \mathtt{Driver}$.

An ontology (also called a *TBox*) is just a collection of such concept inclusions and thus forms a partial order w.r.t. its concepts.

Definition 3.2.3 ($\mathcal{ALC}$-Ontology)

An $\mathcal{ALC}$-ontology $\mathcal{O}$ is a finite set of concept inclusions.

In this case, we have fixed the DL fragment $\mathcal{ALC}$. However, if the fragment is unspecified, we refer to $\mathcal{O}$ as just an *ontology*.

Example 3.2.3

The above introduced knowledge that humans are exactly drivers or pedestrians, crowd members are humans next to other humans, and crowd members are never drivers can be formalized in an $\mathcal{ALC}$-ontology $\mathcal{O}_h = \{\texttt{Human} \equiv \texttt{Driver} \sqcup \texttt{Pedestrian}, \texttt{CrowdMember} \equiv \texttt{Human} \sqcap \exists \texttt{next\_to}.\texttt{Human}, \texttt{CrowdMember} \sqsubseteq \neg \texttt{Driver}\}$.

Besides modeling abstract domain knowledge, DLs also have a data component over which the knowledge is evaluated. Each datum is also called a fact.

Definition 3.2.4 ($\mathcal{ALC}$-Fact)

An $\mathcal{ALC}$-fact is of the form $\texttt{A}(\texttt{a})$ and $\texttt{r}(\texttt{a}, \texttt{b})$ for $\texttt{A} \in \mathsf{N_C}$, $\texttt{a}, \texttt{b} \in \mathsf{N_I}$, and $\texttt{r} \in \mathsf{N_R}$.

Any given number of facts then makes up a database (also called an *ABox*).

Definition 3.2.5 ($\mathcal{ALC}$-Database)

An $\mathcal{ALC}$-database $\mathcal{D}$ is a finite set of facts.

A database hence assigns individuals to concepts and roles. In the following, both facts and concept inclusions are also referred to as *axioms*. We denote the set of individuals that occur in $\mathcal{D}$ by $\mathsf{Ind}(\mathcal{D})$. Finally, an ontology and database together form a KB.

Definition 3.2.6 ($\mathcal{ALC}$-Knowledge Base)

An $\mathcal{ALC}$-Knowledge Base $\mathcal{K}$ is a tuple $(\mathcal{O}, \mathcal{D})$ of an ontology $\mathcal{O}$ and a database $\mathcal{D}$.

Example 3.2.4

For two individuals `Jon` and `Jane`, we create the database $\mathcal{D}_h = \{\texttt{Human}(\texttt{Jon}), \texttt{Human}(\texttt{Jane}), \texttt{next\_to}(\texttt{Jon}, \texttt{Jane}), \texttt{next\_to}(\texttt{Jane}, \texttt{Jon})\}$, stating that both humans are next to each other. Finally, we can use the ontology from Example 3.2.3 to assemble a KB: $\mathcal{K}_h = (\mathcal{O}_h, \mathcal{D}_h)$.

The semantics of ontologies and data is founded in model theory and defined via interpretations. As a first step, we form a basis for defining a semantics on concept, role, and individual names.

Definition 3.2.7 (Interpretation)

An interpretation $\mathcal{I}$ is a tuple $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of a domain $\Delta^{\mathcal{I}}$ and a mapping $\cdot^{\mathcal{I}}$ that assigns a set $\mathtt{A}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ to every $\mathtt{A} \in \mathsf{N_C}$, a binary relation $\mathtt{r}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to every role name $\mathtt{r} \in \mathsf{N_R}$, and an element $\mathtt{a}^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ to every $\mathtt{a} \in \mathsf{N_I}$.

The restriction that if $\mathtt{a} \neq \mathtt{b}$ then $\mathtt{a}^{\mathcal{I}} \neq \mathtt{b}^{\mathcal{I}}$ is called the Unique Name Assumption (UNA) (meaning that if two individuals have distinct names they must be treated as different in the interpretation as well). Throughout this dissertation, we will not make the UNA. Specifically, it is possible that two individuals with different names are represented by the same domain element in an interpretation.

As to incorporate $\mathcal{ALC}$-concept descriptions as well, the interpretation function is extended inductively.

Definition 3.2.8 ($\mathcal{ALC}$-Interpretation)

An $\mathcal{ALC}$-interpretation $\mathcal{I}$ is an interpretation with the interpretation function extended by the following operations on $\mathcal{ALC}$-concepts C and D as well as role names $\mathtt{r}$:

$$
\begin{aligned}
(\neg C)^{\mathcal{I}} &:= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &:= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &:= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\forall \mathtt{r}.C)^{\mathcal{I}} &:= \{c \in \Delta^{\mathcal{I}} \mid \forall d \in \Delta^{\mathcal{I}}.\, (c,d) \in \mathtt{r}^{\mathcal{I}} \rightarrow d \in C^{\mathcal{I}}\} \\
(\exists \mathtt{r}.C)^{\mathcal{I}} &:= \{c \in \Delta^{\mathcal{I}} \mid \exists d \in \Delta^{\mathcal{I}}.\, (c,d) \in \mathtt{r}^{\mathcal{I}} \wedge d \in C^{\mathcal{I}}\}
\end{aligned}
$$

Interpretations form the basis for defining the semantics of $\mathcal{ALC}$-axioms, i.e., concept inclusions and facts.

Definition 3.2.9 (Semantics of $\mathcal{ALC}$-Axioms)

An interpretation $\mathcal{I}$ is a model of a concept inclusion, written $\mathcal{I} \models C \sqsubseteq D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. $\mathcal{I}$ is a model of the facts $\mathtt{A}(\mathtt{a})$ and $\mathtt{r}(\mathtt{a},\mathtt{b})$ if $\mathtt{a}^{\mathcal{I}} \in \mathtt{A}^{\mathcal{I}}$ and $(\mathtt{a}^{\mathcal{I}}, \mathtt{b}^{\mathcal{I}}) \in \mathtt{r}^{\mathcal{I}}$, respectively, for which we write $\mathcal{I} \models \mathtt{A}(\mathtt{a})$ and $\mathcal{I} \models \mathtt{r}(\mathtt{a},\mathtt{b})$.

This definition of semantics for single axioms can now easily be lifted to ontologies, databases, and KBs.

Definition 3.2.10 (Semantics of $\mathcal{ALC}$-Knowledge Bases)

An interpretation $\mathcal{I}$ is a model of an ontology $\mathcal{O}$, written $\mathcal{I} \models \mathcal{O}$, if for all $C \sqsubseteq D \in \mathcal{O}$ it holds that $\mathcal{I} \models C \sqsubseteq D$. $\mathcal{I}$ is a model of a database $\mathcal{D}$, written $\mathcal{I} \models \mathcal{D}$, if for all $\mathtt{A}(\mathtt{a}), \mathtt{r}(\mathtt{a}, \mathtt{b}) \in \mathcal{D}$ it holds that $\mathcal{I} \models \mathtt{A}(\mathtt{a})$ and $\mathcal{I} \models \mathtt{r}(\mathtt{a}, \mathtt{b})$, respectively. Given a Knowledge Base $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, $\mathcal{I}$ is a model of $\mathcal{K}$, denoted by $\mathcal{I} \models \mathcal{K}$, if $\mathcal{I} \models \mathcal{O}$ and $\mathcal{I} \models \mathcal{D}$.

Example 3.2.5

The KB $\mathcal{K}_h$ of Example 3.2.4 has infinitely many models $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, however, all $\Delta^{\mathcal{I}}$ must state that for any $a \in \Delta^{\mathcal{I}}$ with $a \in \mathtt{Jon}^{\mathcal{I}}$ or $a \in \mathtt{Jane}^{\mathcal{I}}$ that $a \in \mathtt{Pedestrian}^{\mathcal{I}}$, i.e., any model asserts both individuals to be pedestrians.

We can leverage these formalisms to define entailment of $\mathcal{ALC}$-axioms α by a KB $\mathcal{K}$ (written $\mathcal{K} \models \alpha$) to hold if all models of $\mathcal{K}$ are also models of α. Note that this semantics implies $\mathcal{ALC}$ to be a monotonic logic, i.e., adding new axioms to the KB does not lead to the invalidation of previously entailed (but not necessarily explicitly asserted) axioms. Formally, monotonicity refers to the fact that if an axiom α is entailed by $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, i.e., $\mathcal{K} \models \alpha$, then for all extensions $\mathcal{K}' = (\mathcal{O}', \mathcal{D}')$ of $\mathcal{K}$ with $\mathcal{O} \subseteq \mathcal{O}'$ and $\mathcal{D} \subseteq \mathcal{D}'$ it holds that $\mathcal{K}' \models \alpha$.

For syntactic convenience, instead of using only concept names $\mathtt{A} \in \mathsf{N_C}$ for facts in the database, we also allow arbitrary concepts to be used in facts, e.g., $(\neg\mathtt{A})(\mathtt{a}) \in \mathcal{D}$. Semantically, any KB $(\mathcal{O}, \mathcal{D})$ with such a complex database fact $C(\mathtt{a}) \in \mathcal{D}$ can be transformed into an equivalent KB, i.e., a KB having the same models, $(\mathcal{O} \cup \{\mathtt{F} \equiv C\}, \mathcal{D} \cup \{\mathtt{F}(\mathtt{a})\})$ that only uses concept names for facts, where $\mathtt{F} \in \mathsf{N_C}$ is a fresh concept name.

Relevant Problems in Description Logics

The literature examines many decision problems for DLs. *Consistency* is one of the central ones, which we present in the following. Consistency intuitively asks whether the axioms in some ontology and database are satisfiable, i.e., if they have a model. If this is not the case, there must be some contradiction within a single axiom or between two or more axioms. Again, all formal definitions are based on the standard reference of Baader et al. [Baa+07].

Definition 3.2.11 (Consistency)

A Knowledge Base $\mathcal{K}$ is consistent if there exists an interpretation $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$.

Example 3.2.6

Adding the fact $\texttt{Driver}(\texttt{Jon})$ to the KB $\mathcal{K}_h$ of Example 3.2.4 makes the KB inconsistent, as $\texttt{Jon}$ is a crowd member, who cannot be drivers.

For $\mathcal{ALC}$-KBs, deciding consistency is ExpTime-complete [Sch91; Fab00]. Tableau methods have established themselves for proving consistency of a KB $\mathcal{K}$ in the area of DLs [Baa+07], where the idea is to create (an abstraction of) a satisfying interpretation $\mathcal{I}$. This constructed model is often called *completion graph* in the literature, and is a directed, labeled graph acting as the central data structure of tableau methods. More specifically, each vertex and edge is labeled with a set of concepts role names, respectively. The data structure is initialized with a vertex for each individual $\texttt{i} \in \mathsf{Ind}(\mathcal{K})$ with label $\{\texttt{i}\} \cup \{\texttt{C} \mid \texttt{C}(\texttt{i}) \in \mathcal{A}\}$, and an edge for each fact $\texttt{r}(\texttt{i}_1, \texttt{i}_2)$ labeled with $\{\texttt{r}\}$. If there is no individual present, we initialize an anonymous individual and label it with $\top$. The graph is iteratively completed during the algorithm by applying so-called *expansion rules* (creating or merging new vertices, labels, or edges in the graph) based on a Negation Normal Form (NNF) representation of the ontology (all negations are pushed inwards using De Morgan's law and duality of $\forall$ and $\exists$). To avoid an infinite expansion of the graph, standard tableau algorithms for $\mathcal{ALC}$ rely on *blocking* of nodes, and thereby creating a finite representation of an infinite interpretation. Here, a node x in the completion graph is blocked by another node y iff it is not an individual and x can be reached from y and the labels of x are contained in the labels of y. Due to nondeterministic choices induced by disjunctions, the tableau algorithm requires nondeterministic branching into two copies of possible completion graphs. The expansion rules for $\mathcal{ALC}$ are:

1. if $C \sqcap D$ is in the label of a non-blocked node and one of C or D are not in its label, add C, D to its label,

2. if $C \sqcup D$ is in the label of a non-blocked node and neither C nor D are in its label, create two branches: one where C and one where D is added to its label,

3. if $\exists \texttt{r}.C$ is in the label of a non-blocked node x and there is no $\texttt{r}$-successor of x with label C, create a fresh node y with label $\{C\}$ and an edge from x to y labeled $\{\texttt{r}\}$,

4. if $\forall \texttt{r}.C$ is in the label of a non-blocked node x and there is an $\texttt{r}$-successor y of x without label C, add C to the label of y, and

5. if $C \sqsubseteq D$ is in the ontology and $\neg C \sqcup D$ is not in the label of a non-blocked node, add $\neg C \sqcup D$ in NNF to its label.

Note that the nondeterminism of the second rule causes an exponential blow-up in run time. These rules are applied to the graph until no rule can be fired anymore.

For deciding whether the generated completion graph represents an abstraction of a model of the KB, one usually checks whether a *clash* is present in the graph, i.e., both C and $\neg C$ exists in some label. The overall procedure returns that $\mathcal{K}$ is consistent iff there is a clash-free completion graph on which no more expansion rules can be fired.

Example 3.2.7

For the KB $\mathcal{K}_h$ of Example 3.2.4, a clash-free completion graph after applying all expansion rules for $\mathcal{ALC}$ is given in Figure 3.1.

$\{\texttt{Human},$
$\neg\texttt{CrowdMember} \sqcup \neg\texttt{Driver},$
$\quad \neg\texttt{Driver},$
$\neg\texttt{CrowdMember} \sqcup (\texttt{Human} \sqcap \exists\texttt{next\_to}.\texttt{Human}),$
$\quad \texttt{Human} \sqcap \exists\texttt{next\_to}.\texttt{Human},$
$\quad\quad \exists\texttt{next\_to}.\texttt{Human},$
$(\neg\texttt{CrowdMember} \sqcup \forall\texttt{next\_to}.\neg\texttt{Human}) \sqcup \texttt{CrowdMember},$
$\quad \forall\texttt{next\_to}.\neg\texttt{Human} \sqcup \texttt{CrowdMember},$
$\quad\quad \texttt{CrowdMember},$
$\neg\texttt{Human} \sqcup (\texttt{Driver} \sqcup \texttt{Pedestrian}),$
$\quad \texttt{Driver} \sqcup \texttt{Pedestrian},$
$\quad\quad \texttt{Pedestrian},$
$(\neg\texttt{Driver} \sqcap \neg\texttt{Pedestrian}) \sqcup \texttt{Human}\}$

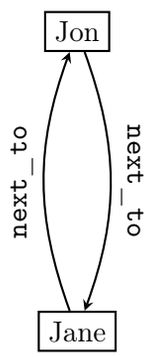

Figure 3.1: Clash-free completion graph for the example KB $\mathcal{K}_h$, where the labels on the left apply to both nodes, with the algorithm's selection of subaxioms indicated by indentation.

No expansion rule is applicable on the completion graph anymore. We can, for example, read from the graph that both `Jon` and `Jane` must be a `Pedestrian`.

When working with DLs, consistency can be a central question to which related problems are reducible. Two of these related problems are:

1. *Subsumption*: A concept D subsumes C in a Knowledge Base $\mathcal{K}$ if for all interpretations $\mathcal{I} \models \mathcal{K}$ it holds that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (written $\mathcal{K} \models C \sqsubseteq D$).

2. *Instance checking*: An individual $\mathtt{a}$ is an instance of a concept C in a Knowledge Base $\mathcal{K}$ if for all interpretations $\mathcal{I} \models \mathcal{K}$ it holds that $\mathtt{a}^{\mathcal{I}} \in C^{\mathcal{I}}$ (written $\mathcal{K} \models C(\mathtt{a})$).

Their reduction to consistency is straightforward [Baa+07]. For subsumption, $\mathcal{K} = (\mathcal{O}, \mathcal{D}) \models C \sqsubseteq D$ iff $(\mathcal{O} \cup \{C \sqcap \neg D\}, \mathcal{D})$ is inconsistent. For instance checking, $\mathcal{K} = (\mathcal{O}, \mathcal{D}) \models C(\mathtt{a})$ iff $(\mathcal{O}, \mathcal{D} \cup \{(\neg C)(\mathtt{a})\})$ is inconsistent.

Moreover, there are function problems over KBs, i.e., problems where the solution space is not binary [Baa+07]. For example, *instance retrieval* asks for all individuals $\mathtt{a}$ satisfying $\mathcal{K} \models C(\mathtt{a})$ and the *realization problem* asks for the most specific concept names $\mathtt{C}$ for which $\mathcal{K} \models \mathtt{C}(\mathtt{a})$ holds.

Expressive Description Logics

Evidently, $\mathcal{ALC}$ is a rather simple DL. To combat this, even more expressive fragments were introduced for increasing expressiveness, especially on roles, while retaining decidability. One of the practically most relevant DLs is $\mathcal{SROIQ}^{(\mathcal{D})}$, as originally defined by Horrocks et al. [HKS06]. It additionally allows for:

- constrained complex role inclusions and assertions on roles (symmetry, asymmetry, transitivity, reflexivity, irreflexivity, and disjointness) ($\mathcal{R}$),
- nominals ($\mathcal{O}$),
- inverse roles ($\mathcal{I}$),
- qualified cardinality restrictions ($\mathcal{Q}$), and
- concrete datatypes ($^{(\mathcal{D})}$).

$\mathcal{S}$ is just an abbreviation for $\mathcal{ALC}$ with transitive roles. We also sometimes refer to $\mathcal{H}$, which restricts $\mathcal{R}$ to only role hierarchies, as, e.g., used in the DL $\mathcal{SHIQ}$. The later presented algorithms for temporal querying operate over $\mathcal{SRIQ}^{(\mathcal{D})}$ KBs and thus do not allow for nominals. For completeness, we still include nominals in the upcoming presentation. Moreover, this will later enable us to understand why nominals can be problematic for the proposed algorithms.

We now introduce the syntax and semantics of these additional operators as presented in the literature [HS01; HKS06]. First, we extend expressiveness on roles. For this, we allow an inverse role for any role in $\mathsf{N_R}$, i.e., we redefine the supply of role names as $\mathsf{N_R}^- = \mathsf{N_R} \cup \{r^- \mid r \in \mathsf{N_R}\}$. We can use both *role assertions* and *role hierarchies*. Role hierarchies are similar to concept inclusions and allow – with some restrictions – modeling composition and inclusion of roles.

Definition 3.2.12 (Role Inclusion)

A role inclusion has the form $\mathtt{r}_0 \circ \cdots \circ \mathtt{r}_k \sqsubseteq \mathtt{s}$ for roles $\mathtt{r}_0, \ldots, \mathtt{r}_k, \mathtt{s} \in \mathsf{N_R}^-$, $k \in \mathbb{N}$.

Example 3.2.8

We can employ role inclusions to model transitivity for spatial roles used to describe

relative positioning in road traffic, e.g., $\texttt{right\_of} \circ \texttt{right\_of} \sqsubseteq \texttt{right\_of}$ (if a is right of b and b is right of c, then a is right of c).

We cannot allow arbitrary role inclusions of the introduced pattern, as cycles in role chains – such as $\texttt{r} \circ \texttt{s} \sqsubseteq \texttt{t}$ and $\texttt{t} \circ \texttt{u} \sqsubseteq \texttt{s}$ – lead to undecidability. To remain decidable, Horrocks et al. introduce the notion of *regular* role inclusions [HS04].

Definition 3.2.13 (Regular Role Inclusions [HS04])

A set of role inclusions is regular if there exists a relation $\prec \subseteq \mathsf{N_R}^- \times \mathsf{N_R}^-$ that is irreflexive and transitive as well as satisfies $\texttt{s} \prec \texttt{r}$ iff $\texttt{s}^- \prec \texttt{r}$, and each role inclusion $\texttt{r}_0 \circ \cdots \circ \texttt{r}_k \sqsubseteq \texttt{s}$ adheres to one of the following constraints:

1. $k = 0$ and $\texttt{r}_0 = \texttt{s}^-$ (symmetry),
2. $k = 1$ and $\texttt{r}_0 = \texttt{r}_1 = \texttt{s}$ (transitivity),
3. $\texttt{r}_i \prec \texttt{s}$ for all $i \in \{0, \ldots, k\}$,
4. $\texttt{r}_0 = \texttt{s}$ and $\texttt{r}_i \prec \texttt{s}$ for all $i \in \{1, \ldots, k\}$, or
5. $\texttt{r}_k = \texttt{s}$ and $\texttt{r}_i \prec \texttt{s}$ for all $i \in \{0, \ldots, k-1\}$.

Note that the constraint of irreflexivity and transitivity effectively forbids cycles in $\prec$. If we also add assertions such as role disjointness, avoiding cycles in role chains is, however, not enough for retaining decidability. For this, we furthermore introduce the concept of *simple* roles.

Definition 3.2.14 (Simple Role [HS04])

For a given set of role inclusions, a role $\texttt{r} \in \mathsf{N_R}^-$ is simple if

1. there is no role inclusion $\texttt{r}_0 \circ \cdots \circ \texttt{r}_k \sqsubseteq \texttt{r}$,
2. there are only role inclusions $\texttt{s} \sqsubseteq \texttt{r}$ with $\texttt{s}$ being a simple role, or
3. $\texttt{r} = \texttt{s}^-$ with $\texttt{s}$ being a simple role.

We also call non-simple roles *complex*. For our example role inclusions $\texttt{r} \circ \texttt{s} \sqsubseteq \texttt{t}$ and $\texttt{t} \circ \texttt{u} \sqsubseteq \texttt{s}$ we find that $\texttt{u}$, $\texttt{r}$, $\texttt{u}^-$, and $\texttt{r}^-$ are simple (as $\texttt{u}$ and $\texttt{r}$ do not occur on the right hand side of a role inclusion) and $\texttt{s}$, $\texttt{t}$, $\texttt{s}^-$, and $\texttt{t}^-$ are complex (as $\texttt{s}$ and $\texttt{t}$ occur on the right hand side of a role inclusion whose left hand side is not a single role). Based on this definition, role assertions allow safely modeling properties of roles, such as symmetry.

Definition 3.2.15 (Role Assertion)

For a role $\mathtt{r} \in \mathsf{N_R}$ and simple roles $\mathtt{s}, \mathtt{s}' \in \mathsf{N_R}$, a role assertion is one of the following expressions: $\mathrm{Sym}(\mathtt{r})$, $\mathrm{Asy}(\mathtt{s})$, $\mathrm{Tra}(\mathtt{r})$, $\mathrm{Ref}(\mathtt{r})$, $\mathrm{Irr}(\mathtt{s})$, or $\mathrm{Dis}(\mathtt{s}, \mathtt{s}')$.

Note that some of these assertions are restricted to simple roles.

Example 3.2.9

The role inclusion of Example 3.2.8 can be abbreviated to $\mathrm{Tra}(\texttt{right\_of})$.

Besides increasing expressiveness of axioms on roles, $\mathcal{SROIQ}^{(\mathcal{D})}$ also extends the concept descriptions from $\mathcal{ALC}$, allowing qualified cardinality restrictions, nominals, and self-reference [HKS06], as well as datatypes [HS01]. We now summarize the existing definitions from the literature. For datatypes, we assume that we have a global, concrete domain Δ_{D} that is disjoint from any interpretation domain, and a supply D of concrete datatypes, including a datatype representing Δ_{D}. Each of these datatypes $d \in \mathsf{D}$ can take values from its domain $d^{\mathsf{D}} \subseteq \Delta_{\mathsf{D}}$. For each datatype, a corresponding datatype representing its negation must exist. To retain decidability, we additionally require that checking for emptiness of intersections of datatypes is decidable [HS01]. For incorporating datatypes into the ontology, they have to be used with a special set of roles $\mathsf{N_D}$, so-called concrete roles, with $\mathsf{N_D} \cap \mathsf{N_R} = \emptyset$.

Definition 3.2.16 ($\mathcal{SROIQ}^{(\mathcal{D})}$-Concept)

An $\mathcal{SROIQ}^{(\mathcal{D})}$-concept C is an $\mathcal{ALC}$-concept C that additionally adheres to the grammar

$$C ::= \leq n\, \mathtt{s}.C \mid \geq n\, \mathtt{s}.C \mid \{\mathtt{a}_0, \ldots, \mathtt{a}_k\} \mid \exists \mathtt{s}.\mathrm{Self} \mid \exists \mathtt{t}.d \mid \forall \mathtt{t}.d$$

where $\mathtt{s} \in \mathsf{N_R}$ and simple, $\mathtt{a}_0, \ldots, \mathtt{a}_k \in \mathsf{N_I}$ with $k \in \mathbb{N}$, $\mathtt{t} \in \mathsf{N_D}$, $d \in \mathsf{D}$, and $n \in \mathbb{N}$.

An $\mathcal{SROIQ}^{(\mathcal{D})}$-ontology now allows both role assertions and inclusions to be added to its ontologies.

Definition 3.2.17 ($\mathcal{SROIQ}^{(\mathcal{D})}$-Ontology)

An $\mathcal{SROIQ}^{(\mathcal{D})}$-ontology $\mathcal{O}$ is a finite set of concept inclusions over $\mathcal{SROIQ}^{(\mathcal{D})}$-concepts, role assertions, and regular role inclusions.

Finally, $\mathcal{SROIQ}^{(\mathcal{D})}$ also extends the expressiveness in the database facts by allowing negated assertions over simple roles.

Definition 3.2.18 ($\mathcal{SROIQ}^{(\mathcal{D})}$-Fact)

An $\mathcal{SROIQ}^{(\mathcal{D})}$-fact is of the form $\mathtt{A}(\mathtt{a})$, $\mathtt{r}(\mathtt{a},\mathtt{b})$, $\neg\mathtt{s}(\mathtt{a},\mathtt{b})$, $\mathtt{t}(\mathtt{a},d)$, or $\neg\mathtt{t}(\mathtt{a},d)$ for $\mathtt{A} \in \mathsf{N_C}$, $\mathtt{a},\mathtt{b} \in \mathsf{N_I}$, $\mathtt{r} \in \mathsf{N_R}$, $\mathtt{s} \in \mathsf{N_R}$ and simple, and $\mathtt{t} \in \mathsf{N_D}$, $d \in \mathsf{D}$.

An $\mathcal{SROIQ}^{(\mathcal{D})}$-KB is syntactically defined analogously to $\mathcal{ALC}$.

Example 3.2.10

The following axioms can extend the $\mathcal{ALC}$ KB $\mathcal{K}_h$ of Example 3.2.4 to an $\mathcal{SROIQ}^{(\mathcal{D})}$ KB:

- $\mathrm{Sym}(\mathtt{next\_to})$ — additional role axioms ($\mathcal{R}$)
- $\mathtt{ValidDrivingTrafficLight} \sqsubseteq \exists \mathtt{has\_color}.\{\mathtt{Green},\mathtt{Yellow}\}$ — nominals ($\mathcal{O}$)
- $\exists \mathtt{drives}.(\forall \mathtt{occupies}^{-}.\mathtt{Passenger}) \sqsubseteq \mathtt{Chauffeur}$ — inverse roles ($\mathcal{I}$)
- $\geq 2\, \mathtt{drives}.\mathtt{Vehicle} \sqsubseteq \mathtt{IllegalDriver}$ — qualified cardinality restrictions ($\mathcal{Q}$)
- $\mathtt{Driver} \sqsubseteq \exists \mathtt{drives}.(\mathtt{Vehicle} \sqcap \exists \mathtt{has\_speed}.\mathbb{Q}_{>0})$, with the datatype $\mathbb{Q}_{>0} = \{x \in \mathbb{Q} \mid x > 0\}$ — concrete datatypes ($^{(\mathcal{D})}$)

We are now ready to define the semantics of $\mathcal{SROIQ}^{(\mathcal{D})}$ based on the works of Horrocks et al. [HS01; HKS06].

Definition 3.2.19 ($\mathcal{SROIQ}^{(\mathcal{D})}$-Interpretation)

An $\mathcal{SROIQ}^{(\mathcal{D})}$-interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is an $\mathcal{ALC}$-interpretation with $\Delta^{\mathcal{I}} \cap \Delta_{\mathsf{D}} = \emptyset$, extended by the following operations on an $\mathcal{SROIQ}^{(\mathcal{D})}$-concept C, role names $\mathtt{r},\mathtt{r}_0,\ldots,\mathtt{r}_j$ with $j \in \mathbb{N}$, individual names $\mathtt{a}_0,\ldots\mathtt{a}_k$ with $k \in \mathbb{N}$, concrete role $\mathtt{t}$, datatype d, and $n \in \mathbb{N}$:

$$
\begin{aligned}
(\geq n\, \mathtt{r}.C)^{\mathcal{I}} &\coloneqq \{c \in \Delta^{\mathcal{I}} \mid |\{d \in \Delta^{\mathcal{I}} \mid (c,d) \in \mathtt{r}^{\mathcal{I}} \wedge \mathtt{d} \in C^{\mathcal{I}}\}| \geq n\} \\
(\leq n\, \mathtt{r}.C)^{\mathcal{I}} &\coloneqq \{c \in \Delta^{\mathcal{I}} \mid |\{d \in \Delta^{\mathcal{I}} \mid (c,d) \in \mathtt{r}^{\mathcal{I}} \wedge \mathtt{d} \in C^{\mathcal{I}}\}| \leq n\} \\
(\{\mathtt{a}_0,\ldots,\mathtt{a}_k\})^{\mathcal{I}} &\coloneqq \{\mathtt{a}_0{}^{\mathcal{I}},\ldots,\mathtt{a}_k{}^{\mathcal{I}}\} \\
(\exists \mathtt{r}.\mathrm{Self})^{\mathcal{I}} &\coloneqq \{c \in \Delta^{\mathcal{I}} \mid (c,c) \in \mathtt{r}^{\mathcal{I}}\} \\
(\mathtt{r}^{-})^{\mathcal{I}} &\coloneqq \{(c,d) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (d,c) \in \mathtt{r}^{\mathcal{I}}\} \\
(\mathtt{r}_0 \circ \cdots \circ \mathtt{r}_j)^{\mathcal{I}} &\coloneqq \mathtt{r}_0{}^{\mathcal{I}} \circ \cdots \circ \mathtt{r}_j{}^{\mathcal{I}} \\
(\forall \mathtt{t}.d) &\coloneqq \{c \in \Delta^{\mathcal{I}} \mid \forall x \in \Delta_{\mathsf{D}}.\, (c,x) \in \mathtt{t}^{\mathcal{I}} \rightarrow x \in d^{\mathsf{D}}\} \\
(\exists \mathtt{t}.d) &\coloneqq \{c \in \Delta^{\mathcal{I}} \mid \exists x \in \Delta_{\mathsf{D}}.\, (c,x) \in \mathtt{t}^{\mathcal{I}} \wedge x \in d^{\mathsf{D}}\}
\end{aligned}
$$

We again use interpretations for defining models of axioms in $\mathcal{SROIQ}^{(\mathcal{D})}$. We assume the already defined semantics of $\mathcal{ALC}$ axioms to also hold for $\mathcal{SROIQ}^{(\mathcal{D})}$.

Definition 3.2.20 (Semantics of $\mathcal{SROIQ}^{(\mathcal{D})}$-Axioms)

Let $\mathcal{I}$ be an $\mathcal{SROIQ}^{(\mathcal{D})}$-interpretation. The semantics of $\mathcal{I}$ on $\mathcal{SROIQ}^{(\mathcal{D})}$-axioms is, additional to the semantics of $\mathcal{ALC}$-axioms, defined as:

- $\mathcal{I} \models \neg\mathtt{r}(\mathtt{a},\mathtt{b})$ if $(\mathtt{a}^{\mathcal{I}},\mathtt{b}^{\mathcal{I}}) \notin \mathtt{r}^{\mathcal{I}}$.
- $\mathcal{I} \models \mathtt{r_0} \circ \cdots \circ \mathtt{r_k} \sqsubseteq \mathtt{s}$ if $(\mathtt{r_0} \circ \cdots \circ \mathtt{r_k})^{\mathcal{I}} \subseteq \mathtt{s}^{\mathcal{I}}$.
- $\mathcal{I} \models \mathrm{Sym}(\mathtt{r})$ if $(c,d) \in \mathtt{r}^{\mathcal{I}}$ implies $(d,c) \in \mathtt{r}^{\mathcal{I}}$.
- $\mathcal{I} \models \mathrm{Asy}(\mathtt{r})$ if $(c,d) \in \mathtt{r}^{\mathcal{I}}$ implies $(d,c) \notin \mathtt{r}^{\mathcal{I}}$.
- $\mathcal{I} \models \mathrm{Tra}(\mathtt{r})$ if $(c,d) \in \mathtt{r}^{\mathcal{I}}$ and $(d,e) \in \mathtt{r}^{\mathcal{I}}$ implies $(c,e) \in \mathtt{r}^{\mathcal{I}}$.
- $\mathcal{I} \models \mathrm{Ref}(\mathtt{r})$ if $(d,d) \in \mathtt{r}^{\mathcal{I}}$ for all $d \in \Delta^{\mathcal{I}}$.
- $\mathcal{I} \models \mathrm{Irr}(\mathtt{r})$ if $(d,d) \notin \mathtt{r}^{\mathcal{I}}$ for all $d \in \Delta^{\mathcal{I}}$.
- $\mathcal{I} \models \mathrm{Dis}(\mathtt{r},\mathtt{s})$ if $\mathtt{r}^{\mathcal{I}} \cap \mathtt{r}^{\mathcal{I}} = \emptyset$.

The semantics of $\mathcal{SROIQ}^{(\mathcal{D})}$-KBs is equivalent to the case of $\mathcal{ALC}$-KBs, with the extension that an interpretation $\mathcal{I}$ models a database if also $\mathcal{I} \models \neg\mathtt{r}(\mathtt{a},\mathtt{b})$ holds for all negated role assertions in the database.

Allowing such expressive constructs in a DL results in increased complexity of the decision procedures. More specifically, deciding consistency of an $\mathcal{SROIQ}$-KB is N2ExpTime-complete [Kaz08]. Expressiveness does not only influence complexity, but also leads to loss of (desirable) properties of the DL. For example, $\mathcal{ALC}$ has the finite-model property, i.e., if an $\mathcal{ALC}$-KB has a model, there is also a finite model. For $\mathcal{SROIQ}^{(\mathcal{D})}$, however, this property is lost, and not even a tree-shape of the models can be guaranteed [Tob00].

Description Logics and OWL2 DL

The introduced DL fragment $\mathcal{SROIQ}^{(\mathcal{D})}$ has the same expressiveness as OWL2 DL, a widely used ontology engineering language. In Chapter 5, we use OWL2 DL to evaluate our algorithms, and thus a short introduction is in order. OWL2 DL, as a successor of the first Web Ontology Language, has initially been published as a World Wide Web Consortium (W3C) Recommendation in 2009 and its current stable version is dated to 2012. While it supports various syntaxes, such as an XML serialization of an RDF graph, this work relies solely on the so-called functional syntax (a prefix notation). All core axioms in OWL2 DL functional-style syntax can be translated to an equivalent in $\mathcal{SROIQ}^{(\mathcal{D})}$, as

shown in Table 3.1, and hence no new features are introduced here. Note that OWL2 DL allows for various additional syntax that is syntactic sugar to the operators described in Table 3.1. This includes, e.g., the expression `EquivalentClasses(A B)` which just stands for `SubClassOf(A B)` and `SubClassOf(B A)`.

OWL2 is used as an input language by reasoners, software tools that perform the reasoning tasks introduced above. Many major reasoners support OWL2 DL, which is equivalent to the full expressiveness of $\mathcal{SROIQ}^{(\mathcal{D})}$. Other fragments (both less and more expressive), such as OWL2 EL and OWL2 Full, are included in OWL2, too.

Table 3.1: A syntactical translation between $\mathcal{SROIQ}^{(\mathcal{D})}$ and OWL2 DL.

$\mathcal{SROIQ}^{(\mathcal{D})}$	OWL2 DL
Axioms	
$\mathtt{A}(\mathtt{b})$	`ClassAssertion(A b)`
$\mathtt{r}(\mathtt{a},\mathtt{b})$	`ObjectPropertyAssertion(r a b)`
$\neg\mathtt{r}(\mathtt{a},\mathtt{b})$	`NegativeObjectPropertyAssertion(r a b)`
$\mathtt{t}(\mathtt{a},d)$	`DataPropertyAssertion(t a d)`
$\neg\mathtt{t}(\mathtt{a},d)$	`NegativeDataPropertyAssertion(t a d)`
$A \sqsubseteq B$	`SubClassOf(A B)`
$\mathtt{r}_0 \circ \cdots \circ \mathtt{r}_k \sqsubseteq \mathtt{s}$	`SubObjectPropertyOf(ObjectPropertyChain(`$\mathtt{r}_0$ `...`$\mathtt{r}_k$`) s)`
$\mathrm{Sym}(\mathtt{r})$	`SymmetricObjectProperty(r)`
$\mathrm{Asy}(\mathtt{r})$	`AsymmetricObjectProperty(r)`
$\mathrm{Tra}(\mathtt{r})$	`TransitiveObjectProperty(r)`
$\mathrm{Ref}(\mathtt{r})$	`ReflexiveObjectProperty(r)`
$\mathrm{Irr}(\mathtt{r})$	`IrreflexiveObjectProperty(r)`
$\mathrm{Dis}(\mathtt{r},\mathtt{s})$	`DisjointObjectProperties(r s)`
Concept Descriptions	
$\neg A$	`ObjectComplementOf(A)`
$A \sqcap B$	`ObjectIntersectionOf(A B)`
$A \sqcup B$	`ObjectUnionOf(A B)`
$\forall\mathtt{r}.A$	`ObjectAllValuesFrom(r A)`
$\exists\mathtt{r}.A$	`ObjectSomeValuesFrom(r A)`
$\leq n\,\mathtt{r}.A$	`ObjectMaxCardinality(n r A)`
$\geq n\,\mathtt{r}.A$	`ObjectMinCardinality(n r A)`
$\{\mathtt{a}_0,\ldots,\mathtt{a}_k\}$	`ObjectOneOf(`$\mathtt{a}_0$ `...`$\mathtt{a}_k$`)`
$\exists\mathtt{r}.\mathrm{Self}$	`ObjectHasSelf(s)`
$\exists\mathtt{t}.d$	`DataSomeValuesFrom(t d)`
$\forall\mathtt{t}.d$	`DataAllValuesFrom(t d)`

3.2.2 Conjunctive Query Answering over Description Logics

We have already introduced various problems over DLs. In this dissertation, the central problem of interest over DLs is *query answering*. Specifically CQ answering has been of interest to the research community due to its balance between applicability and performance [Pog+08]. As they will be the foundation of the novel temporal query language proposed in Section 3.4, we now take a deeper look into existing literature and standard definitions for CQs over expressive DL KBs. This subsection compiles a curated overview of techniques established in the literature, which is imperative for understanding our results presented in Chapter 4 and Chapter 5.

Formal Foundations of Conjunctive Query Answering

In a nutshell, CQs are equivalent to the SELECT-FROM-WHERE-fragment of SQL and allow asking for individuals in the database that fulfill a conjunct of concept and role constraints. A standard way of defining CQs is to view them as a restricted variant of first-order logic formulae with free (answer) variables [Baa+07, Chapter 16].

Definition 3.2.21 (Conjunctive Query)

A Conjunctive Query over supplies of variable names $\mathsf{N_V}$, individual names $\mathsf{N_I}$, concept names $\mathsf{N_C}$, role names $\mathsf{N_R}$, concrete roles $\mathsf{N_D}$, and datatypes D adheres to the grammar

$$\begin{aligned}\varphi &::= \exists \xi.\psi \mid \psi\\ \xi &::= \xi, \xi \mid v\\ \psi &::= \psi \wedge \psi \mid \alpha\\ \alpha &::= \mathtt{C}(x) \mid \mathtt{r}(x,x) \mid \mathtt{t}(x,d)\end{aligned}$$

with $v \in \mathsf{N_V}$, $x \in \mathsf{N_I} \cup \mathsf{N_V}$, $d \in \mathsf{D}$, $\mathtt{C} \in \mathsf{N_C}$, $\mathtt{r} \in \mathsf{N_R}$, and $\mathtt{t} \in \mathsf{N_D}$. We call α a *query atom* and the set of all atoms $\mathsf{Atoms}(\varphi)$. The *quantified variables* in φ are denoted by $\mathsf{QVar}(\varphi)$. The *answer variables* in φ, i.e., all $x \in \mathsf{N_V}$ that occur in φ but are not in $\mathsf{QVar}(\varphi)$, are denoted by $\mathsf{AVar}(\varphi)$. To denote a query over answer variables, we also write $\varphi(\vec{x})$. If $\mathsf{AVar}(\varphi) = \emptyset$, we call φ *Boolean*.

We sometimes make the answer variables explicit by denoting a non-Boolean query as $\varphi(\vec{x})$, where $\vec{x}$ is an ordered version of $\mathsf{AVar}(\varphi)$. We introduce the special sets $\mathsf{Ind}(\varphi) \subseteq \mathsf{N_I}$ to denote the individuals in φ and $\mathsf{Var}(\varphi) := \mathsf{QVar}(\varphi) \cup \mathsf{AVar}(\varphi)$. To avoid confusion between variable and individual names, we use the convention that $\mathsf{Var}(\varphi) \cap \mathsf{Ind}(\varphi) = \emptyset$ and $\mathsf{QVar}(\varphi) \cap \mathsf{AVar}(\varphi) = \emptyset$.

Example 3.2.11

The non-Boolean CQ $\exists y.\mathtt{Driver}(x) \wedge \mathtt{next\_to}(x,y) \wedge \mathtt{Human}(y)$ has one quantified

variable (y) and one answer variable (x), and asks for all drivers that are next to some human. It does not contain references to individuals.

The semantics of Boolean CQs is defined in terms of matches into interpretations, as usually done in the literature [HT00; Gli+08; CGL08].

Definition 3.2.22 (Semantics of Boolean Conjunctive Queries)

For a Boolean Conjunctive Query φ and an interpretation $\mathcal{I}$, $\mathcal{I} \models \varphi$ iff there exists a function $\pi\colon \mathsf{Var}(\varphi) \cup \mathsf{Ind}(\varphi) \to \Delta^{\mathcal{I}}$ with

1. $\pi(\mathtt{a}) = \mathtt{a}^{\mathcal{I}}$ for all $\mathtt{a} \in \mathsf{Ind}(\varphi)$,
2. $\pi(x) \in \mathtt{C}^{\mathcal{I}}$ for all atoms of the form $\mathtt{C}(x)$ in φ, and
3. $(\pi(x), \pi(x')) \in \mathtt{r}^{\mathcal{I}}$ for all atoms of the form $\mathtt{r}(x, x')$ in φ.

Hence, an interpretation satisfies a Boolean CQ if the interpretation can respect its constraints.

We consider two standard problems for CQs based on the above definition. The first is CQ entailment, a well-examined decision problem asking whether a given KB guarantees the query to hold [Gli+08].

Definition 3.2.23 (Conjunctive Query Entailment)

A Knowledge Base $\mathcal{K}$ entails a Boolean Conjunctive Query φ, written $\mathcal{K} \models \varphi$, if for all interpretations $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$ it holds that $\mathcal{I} \models \varphi$.

Note this definition works in any fragment of DLs, as long $\mathcal{I} \models \mathcal{K}$ is well-defined. Deciding CQ entailment can be reduced to consistency, for which we will later show a standard procedure from the literature for a large subclass of CQs.

Example 3.2.12

We have already seen that in the example KB $\mathcal{K}_h$ from Example 3.2.4 both `Jane` and `Jon` must be pedestrians, i.e., $\mathcal{K}_h \models \mathtt{Pedestrian}(\mathtt{Jane}) \wedge \mathtt{Pedestrian}(\mathtt{Jon})$, as in all models of $\mathcal{K}_h$ it is guaranteed that both individuals are pedestrians: crowd members cannot be drivers and thus both must be pedestrians. For the models $\mathcal{I}$ of $\mathcal{K}_h$, the implied matching function π (cf. Definition 3.2.22) then satisfies $\pi(\mathtt{Jane}) = \mathtt{Jane}^{\mathcal{I}} \in \mathtt{Pedestrian}^{\mathcal{I}}$ and $\pi(\mathtt{Jon}) = \mathtt{Jon}^{\mathcal{I}} \in \mathtt{Pedestrian}^{\mathcal{I}}$.

In practice, however, we are concerned not only with Boolean CQs but want to fetch answers to non-Boolean queries, similar to the standard database setting as, e.g., in an SQL query. For this, the CQ answering problem over DL ontologies has been introduced and thoroughly examined [Gli+08; CGL08].

Definition 3.2.24 (Conjunctive Query Answering)

For a Knowledge Base $\mathcal{K}$ and a Conjunctive Query $\varphi(\vec{x})$, the Conjunctive Query answering problem asks to compute the set of certain answers $\mathsf{cert}_{\mathcal{K}}(\varphi(\vec{x})) = \{\vec{a} \in \mathsf{Ind}(\mathcal{D})^{|\vec{x}|} \mid \mathcal{K} \models \varphi(\vec{a})\}$, where $\varphi(\vec{a})$ denotes the Boolean Conjunctive Query where each variable x_i in $\vec{x}$ has been replaced by its corresponding individual a_i in $\vec{a}$.

Note that the special case of answering a Boolean CQ φ is handled in above definition by using the empty answer to represent entailment, i.e., $\emptyset \in \mathsf{cert}_{\mathcal{K}}(\varphi)$ iff $\mathcal{K} \models \varphi$. Naïvely, CQ answering can be (exponentially) reduced to CQ entailment by just iterating through all possible candidates in $\mathsf{Ind}(\mathcal{K})^{|\vec{x}|}$.

Example 3.2.13

For the example KB $\mathcal{K}_h$ of Example 3.2.4, we can use non-Boolean CQs to ask for the set of certain answers in the data under the axioms of $\mathcal{O}_h$. Take the (very simple) CQ $\varphi_1(x) = \texttt{Pedestrian}(x)$, which has $\mathsf{cert}_{\mathcal{K}_h}(\varphi_1(x)) = \{\texttt{Jane}, \texttt{Jon}\}$. Existentially quantified variables allow us to refer to anonymous individuals implied by the ontology, e.g., asking for any driver that is next to another human using the CQ $\varphi_2(x) = \exists y.\texttt{Driver}(x) \wedge \texttt{next\_to}(x, y) \wedge \texttt{Human}(y)$ (which is the CQ already introduced in Example 3.2.11). However, this query never has a certain answer on $\mathcal{K}_h$, as drivers are humans, but humans next to other humans are crowd members, which in turn cannot be drivers. Hence, the anonymous individual y cannot exist.

CQ entailment has shown to be decidable for $\mathcal{SROIQ}$ (given that the CQ contains only simple roles). This result holds even for UCQs, a generalization of CQs that allows for disjunctions of CQs [RG10]. The computational complexity of CQ answering ranges from ExpTime-completeness for $\mathcal{ALC}$ up to 2ExpTime-completeness for DLs between $\mathcal{ALCI}$ and $\mathcal{SHIQ}$ [Lut08].

Answering is often implemented only for tree-shaped Conjunctive Queries (tCQs). The usual definition in the literature does not allow such queries to have, w.r.t. the quantified variables, cycles in their roles (regardless of direction) and not more than one incoming role per quantified variable [HT00; Gli+08].

Definition 3.2.25 (Tree-shaped Conjunctive Query)

A Conjunctive Query φ is called *tree-shaped* if for all $y \in \mathsf{QVar}(\varphi)$

- it holds that $|\{z \in \mathsf{QVar}(\varphi) \mid \mathtt{r}(z, y) \in \mathsf{Atoms}(\varphi)\}| \leq 1$, i.e., there are no colliders, and
- there is no sequence $(y, y_0)(y_0, y_1) \dots (y_{k-1}, y_k)(y_k, y)$ with $y_i, y_{i+1} \in \mathsf{QVar}(\varphi)$, and $\mathtt{r}(y_i, y_{i+1}) \in \mathsf{Atoms}(\varphi)$ or $\mathtt{r}(y_{i+1}, y_i) \in \mathsf{Atoms}(\varphi)$ for $i \in \{0, \dots, k-1\}$, i.e., there are no undirected cycles.

All CQs given in the above examples are tree-shaped. With this subclass of CQs, the established rolling-up technique becomes applicable and the query can effectively be represented by a single axiom. This has been leveraged to reduce query answering to instance retrieval [HT00]. While theoretical considerations of full query answering have been made in expressive DLs, we are currently not aware of an implementation. On the other hand, the practical benefits gained by reducing query answering to instance retrieval lead to the ability of offering highly optimized algorithms [Sir06], and we hence focus on tree-shaped CQs in the following.

Deciding Tree-Shaped Conjunctive Query Entailment

We now describe the established procedure for checking entailment of a Boolean tCQ φ via rolling up [HT00], as this efficient algorithm forms the foundation of our later presented novel temporal query answering procedures. From now on, we assume that $\mathcal{K}$ is consistent, which can be checked by the previously described tableau method. Otherwise, all queries are trivially entailed. In the simplest case, we have $\varphi = \mathtt{C}(\mathtt{a})$, i.e., we are confronted with the already introduced instance check. Then, $\mathcal{K} = (\mathcal{O}, \mathcal{D}) \models \varphi$ iff $(\mathcal{O}, \mathcal{D} \cup \{\neg\mathtt{C}(\mathtt{a})\})$ is inconsistent. This holds because of the following equivalences:

$$
\begin{aligned}
&(\mathcal{O}, \mathcal{D} \cup \{\neg\mathtt{C}(\mathtt{a})\}) \text{ is inconsistent iff} \\
&\text{there is no interpretation } \mathcal{I} \text{ s.t. } \mathcal{I} \models \mathcal{K} \text{ and } \mathcal{I} \models \neg\mathtt{C}(\mathtt{a}) \text{ iff} \\
&\text{for all interpretations } \mathcal{I} \text{ it holds that } \mathcal{I} \not\models \mathcal{K} \text{ or } \mathcal{I} \models \mathtt{C}(\mathtt{a}) \text{ iff} \\
&\text{for all interpretations } \mathcal{I} \text{ it holds that if } \mathcal{I} \models \mathcal{K} \text{ then } \mathcal{I} \models \mathtt{C}(\mathtt{a}),
\end{aligned}
$$

which is the definition of CQ entailment. A similar pattern can be applied to existentially quantified variables: Here, $\mathcal{K} = (\mathcal{O}, \mathcal{D}) \models \exists y.\mathtt{C}(y)$ iff $(\mathcal{O} \cup \{\top \sqsubseteq \neg\mathtt{C}\}, \mathcal{D})$ is inconsistent.

Therefore, if we are able to transform the tCQ into a single atom of the form $\mathtt{C}(v)$ with v being an answer of existentially quantified variable, we can check entailment easily by a consistency check. The rolling up procedure, described by Horrocks and Tessaris, can

be used for this transformation [HT00]. Here, we interpret the query as a graph, where vertices are individuals and variables, labeled with their concept constraints, and edges are roles between them.

Definition 3.2.26 (Query Graph)

For a Boolean Conjunctive Query φ, the (labeled) query graph is defined as $G_\varphi \coloneqq (V_\varphi, E_\varphi, L_\varphi)$ with $V_\varphi \coloneqq \mathsf{Ind}(\varphi) \cup \mathsf{Var}(\varphi)$, $E_\varphi \coloneqq \{(x, \mathtt{r}, y) \mid \mathtt{r}(x, y) \in \mathsf{Atoms}(\varphi)\}$, and, for $\mathsf{Atoms}(v) \coloneqq \{\alpha \in \mathsf{Atoms}(\varphi) \mid \text{there is some } \mathtt{C} \in \mathsf{N_C}.\ \alpha = \mathtt{C}(v)\}$,

$$L_\varphi(v) \coloneqq \begin{cases} \bigsqcap_{\mathtt{C}(v) \in \mathsf{Atoms}(v)} \mathtt{C} \sqcap \{v\} & \text{if } v \in \mathsf{Ind}(\varphi) \text{ and not root and } \mathsf{Atoms}(v) \neq \emptyset, \\ \top \sqcap \{v\} & \text{if } v \in \mathsf{Ind}(\varphi) \text{ and not root and } \mathsf{Atoms}(v) = \emptyset, \\ \bigsqcap_{\mathtt{C}(v) \in \mathsf{Atoms}(v)} \mathtt{C} & \text{if } (v \notin \mathsf{Ind}(\varphi) \text{ or root}) \text{ and } \mathsf{Atoms}(v) \neq \emptyset, \\ \top & \text{if } (v \notin \mathsf{Ind}(\varphi) \text{ or root}) \text{ and } \mathsf{Atoms}(v) = \emptyset. \end{cases}$$

We hence label the nodes according to their concepts appearing in the query (if there any, otherwise we use $\top$). If the atom refers to an individual, we put that constraint on the label as well – except for the root node, for which the possible individual will be handled separately later on.

L_φ requires nominals to be available in the DL. If the DL under consideration does not support that, we can equivalently use a fresh concept $\mathtt{P_v} \in \mathsf{N_C}$ in place of $\{v\}$ and additionally assert $\mathtt{P_v}(v) \in \mathcal{D}$ [HT00]. Moreover, in G_φ, we call any $v \in V_\varphi$ with $\{v' \in V_\varphi \mid (v, \mathtt{r}, v') \in E_\varphi\} = \emptyset$ and $|\{v' \in V_\varphi \mid (v', \mathtt{r}, v) \in E_\varphi\}| \geq 1$ a leaf.

In what follows, we assume w.l.o.g. that we have a single root node, i.e., one tree, in a non-empty graph. If the query contains multiple trees w.r.t. its roles, it can just be split up and examined separately. The algorithm RollUp(G_φ) for rolling up a tCQ φ is defined recursively. For this, we first define the label that is created by rolling a single edge as RollUpEdge($(v, \mathtt{r}, v')$) $\coloneqq L_\varphi(v) \sqcap \exists \mathtt{r}.(L_\varphi(v'))$. For the base case $E_\varphi = \{(v, \mathtt{r}, v')\}$, RollUp($G_\varphi$) returns the concept RollUpEdge(e). Otherwise, there are two possibilities:

1. If G_φ has a leaf $v \in V_\varphi$, define $E_R \coloneqq \{(v', \mathtt{r}, v) \in E_\varphi\}$.
2. Else, G_φ must have a cycle $(v_0, \mathtt{r_0}, v_1)(v_1, \mathtt{r_1}, v_2) \dots (v_{k-1}, \mathtt{r_{k-1}}, v_k)(v_k, \mathtt{r_k}, v_0)$ with some $i \in \{0, \dots, k\}$ for which $v_i \notin \mathsf{QVar}(\varphi)$. Then, define $E_R \coloneqq \{(v', \mathtt{r}, v_i) \in E_\varphi\}$.

The procedure returns RollUp($(V_\varphi, E_\varphi \setminus E_R, L'_\varphi)$) with

$$L'_\varphi(v) \coloneqq \begin{cases} \text{RollUpEdge}((v', \mathtt{r}, v)) & \text{for all } (v', \mathtt{r}, v) \in E_R, \\ L_\varphi(v) & \text{otherwise.} \end{cases}$$

Once the concept RollUp(G_φ) $= C$ is computed, we need to transform the given KB $\mathcal{K}$ into a KB $\mathcal{K}'$ s.t. $\mathcal{K} \models \varphi$ iff $\mathcal{K}'$ is consistent. For this, we distinguish two cases; the single root node v either refers to an existentially quantified variable or to an individual:

1. If $v \in \mathsf{QVar}(\varphi)$, then $\mathcal{K}' \coloneqq (\mathcal{O} \cup \{\top \sqsubseteq \neg C\}, \mathcal{D})$.

2. If $v \in \mathsf{Ind}(\varphi)$, then $\mathcal{K}' \coloneqq (\mathcal{O}, \mathcal{D} \cup \{(\neg C)(v)\})$.

Example 3.2.14

Consider the CQ $\exists x, y, z.\texttt{Pedestrian}(\texttt{Jane}) \land \texttt{next\_to}(\texttt{Jane}, x) \land \texttt{Human}(x) \land \texttt{next\_to}(\texttt{Jane}, y) \land \texttt{next\_to}(y, z)$. The initial query graph and subsequent roll up steps are depicted in Figure 3.2.

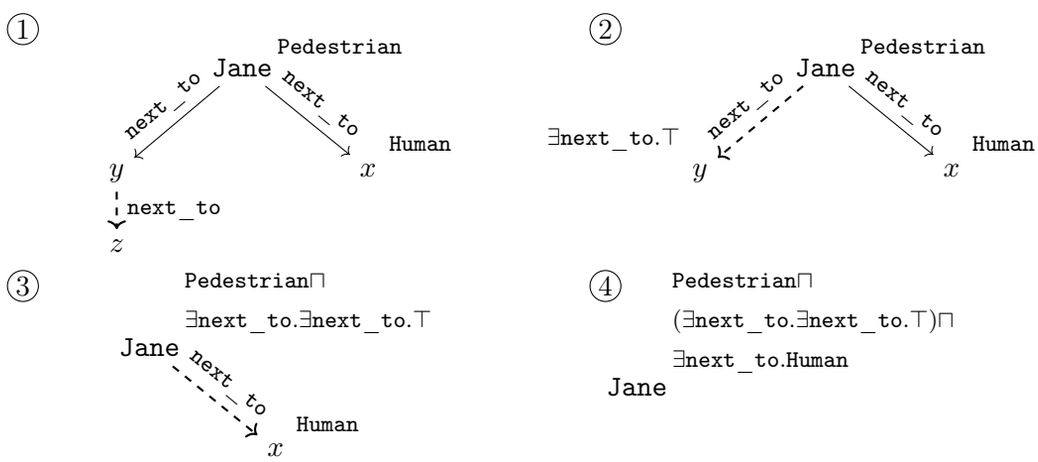

Figure 3.2: Step-by-step rolling up of $\exists x, y, z.\texttt{Pedestrian}(\texttt{Jane}) \land \texttt{next\_to}(\texttt{Jane}, x) \land \texttt{Human}(x) \land \texttt{next\_to}\ (\texttt{Jane}, y) \land \texttt{next\_to}(y, z)$, with the selected edge to roll up in the current step being dashed.

This leads to an instance check of `Jane` for the concept $C = \texttt{Pedestrian} \sqcap (\exists \texttt{next\_to}.\ \exists \texttt{next\_to}.\top) \sqcap \exists \texttt{next\_to}.\texttt{Human}$, which can be achieved on our example KB $\mathcal{K}_h$ by checking consistency of $(\mathcal{O}_h, \mathcal{D}_h \cup \{\neg C(\texttt{Jane})\})$. If the KB is inconsistent – which is the case as `Jane` is a pedestrian and next to the human `Jon` who again is next to `Jane` – the Boolean query is entailed.

Horrocks and Tessaris note that the tree-shape constraint on CQs can be dropped for DLs exhibiting the tree model property, i.e., if the KB has a model it also has a model that is a tree w.r.t. the interpretation of roles. Leveraging the restrictiveness of role assertions, cycles and colliders in existentially quantified variables can then be broken up by replacing one of the quantified variables nondeterministically with an individual [HT00]. Many expressive DLs, such as $\mathcal{ALC}$ and $\mathcal{SRIQ}$, exhibit the tree model property [BBP17]. In contrast, $\mathcal{SROIQ}$ does not have the tree model property [Tob00].

Tree-Shaped Conjunctive Query Answering in Practice

The above summarized rolling-up technique provides a sound and complete procedure for checking entailment of Boolean tCQs, which can be directly leveraged for query answering: Simply iterate through all answer candidates and check entailment for each resulting Boolean tCQ by rolling it up. However, this can be inefficient as there are exponentially many answer candidates. Therefore, state-of-the-art reasoners use techniques to avoid consistency checks of rolled-up Boolean tCQs as much as possible. We now describe the major practical optimizations that were leveraged for an efficient implementation by the contributors of the reasoner Pellet [SP06; KK11]. Again, we present only established results from the literature. Specifically, it was found that it is not required to use rolling-up in case of a CQ not containing an existentially quantified variable. And even in queries with such variables, the set of possible candidates to iterate through can be – in practice – drastically decreased.

Let us first consider the case without any quantified variables, as described by Sirin and Parsia [SP06]. For this, the authors note that it is comparatively cheap to compute instances of the named concepts and roles used in the query – which is the previously introduced task of instance retrieval. It sometimes can be done even without an additional consistency check. Details on state-of-the-art methods to efficiently compute instances are presented below, but let us first assume that such a tractable method exists. Then, a query of the form $\varphi(x_0, x_1) = \mathtt{C}(x_0) \wedge \mathtt{r}(x_0, x_1) \wedge \mathtt{D}(x_1)$ can, in theory, be examined as follows:

1. compute $\mathsf{Ind}(\mathtt{C}, \mathcal{K}) = \{\mathtt{a} \in \mathsf{Ind}(\mathcal{K}) \mid \mathcal{K} \models \mathtt{C}(\mathtt{a})\}$,
2. compute $\mathsf{Ind}(\mathtt{r}, \mathcal{K}) = \{(\mathtt{a}, \mathtt{b}) \in \mathsf{Ind}(\mathcal{K})^2 \mid \mathcal{K} \models \mathtt{r}(\mathtt{a}, \mathtt{b})\}$,
3. compute $\mathsf{Ind}(\mathtt{D}, \mathcal{K}) = \{\mathtt{a} \in \mathsf{Ind}(\mathcal{K}) \mid \mathcal{K} \models \mathtt{D}(\mathtt{a})\}$, and
4. return $\{(\mathtt{a}, \mathtt{b}) \in \mathsf{Ind}(\mathtt{r}, \mathcal{K}) \mid \mathtt{a} \in \mathsf{Ind}(\mathtt{C}, \mathcal{K}) \wedge \mathtt{b} \in \mathsf{Ind}(\mathtt{D}, \mathcal{K})\}$.

With the naïve technique rolling up and iterating over all answer candidates, a large amount of consistency checks has to be performed ($\mathsf{Ind}(\mathcal{K})^2$ many). With the above sketched optimization, it is possible to reduce the number of consistency checks to those that are required for instance retrieval, which, due to their practical optimizations outlined below, are significantly fewer.

Note that the above procedure answers the query atom by atom. For atom number i, it does not use knowledge gained when answering atoms 0 to $i - 1$. In practice, the sets returned by instance retrieval are rather small, which we can exploit for further optimization. Therefore, after computing $\mathsf{Ind}(\mathtt{C}, \mathcal{K})$, it can be more efficient to not compute $\mathsf{Ind}(\mathtt{r}, \mathcal{K})$ over the whole KB, but rather, for the comparatively few $\mathtt{i_0} \in \mathsf{Ind}(\mathtt{C}, \mathcal{K})$, the set $\mathsf{Ind}(\exists \mathtt{r}^-.\{\mathtt{i_0}\}, \mathcal{K})$. This also leaves us with even fewer candidates for x_1, namely those

who have an $\mathtt{r}$-predecessor in $\mathtt{C}$. Therefore, instead of computing $\mathsf{Ind}(\mathtt{D}, \mathcal{K})$, we just check for which candidates $\mathtt{i_1}$ it holds that $\mathcal{K} \models \mathtt{D}(\mathtt{i_1})$.

The above sketched procedure relies, however, on efficient methods to fetch $\mathsf{Ind}(C, \mathcal{K})$ for some concept C. For this, binary instance retrieval (in contrast to the brute-force linear version) has been proposed in the literature [MHW06], which we now summarize shortly. Here, the basic idea is to check whether a *set* of candidates Can (instead of a single individual) is guaranteed to not be within the concept C for some KB $\mathcal{K} = (\mathcal{O}, \mathcal{D})$. This can be done by checking consistency of $(\mathcal{O}, \mathcal{D} \cup \{(\neg C)(\mathtt{c}) \mid \mathtt{c} \in \mathsf{Can}\})$, which holds iff $(\mathcal{O}, \mathcal{D}) \not\models C(\mathtt{c})$ for all $\mathtt{c} \in \mathsf{Can}$. If this is the case, we can exclude the set. Otherwise, there must be some member of C in Can, and we split in two parts – similar to binary search. The same procedure is applied to both parts until $|\mathsf{Can}| = 1$. Typically, we start with the set of all individuals $\mathsf{Ind}(\mathcal{K})$, and ideally exclude those for which membership can be guaranteed a-priori. Such exclusions have high potential for improving efficiency of binary instance retrieval, as they reduce the chance of encountering members in a partition (and thus, having to partition again). The basic idea is to inspect the completion graph generated by the initial consistency check. If the graph labels an individual $\mathtt{i}$ with C and the label C does not come from a nondeterministic choice during expansion or all other nondeterministic choices were examined and closed with a clash, then we can infer that this individual must belong to the concept in all models. The idea of statically inferring membership can be extended by more involved syntactic analyses [Sir06]. Note that even with binary instance retrieval, if Can contains at least one member of C, one will have to check $\mathcal{K} \models C(\mathtt{i})$ eventually. Here, the pseudo model merging technique can be employed to reject $\mathcal{K} \models C(\mathtt{i})$ for non-members of C without checking consistency [SP06; HMT01].

By using the atom-by-atom approach sketched above, Sirin remarks that we often encounter fetching individuals for roles of the form $\mathtt{r}(x, \mathtt{i})$, $\mathtt{r}(\mathtt{i}, x)$, or $\mathtt{r}(x, y)$ [Sir06]. Hence, we have the special case to retrieve individuals for the concept $C = \exists \mathtt{r}.\{\mathtt{i}\}$, $C = \exists \mathtt{r}^-.\{\mathtt{i}\}$, or both, which can also be optimized. The basic idea is similar to checking the membership of a concept by using the generated completion graph. Let us assume we examine $\mathtt{r}(x, \mathtt{i})$. We then have three cases: If the completion graph asserts the role $\mathtt{r}$ between an individual $\mathtt{a}$ and $\mathtt{i}$ and no nondeterminism lead to this fact in the completion graph, $\mathtt{a}$ can be returned with certainty. If, however, some nondeterminism lead to this fact in the completion graph, we must confirm $\mathcal{K} \models \mathtt{r}(\mathtt{a}, \mathtt{i})$ with a consistency check ($\mathtt{a}$ is thus a potential candidate). Lastly, if there is no such edge in the completion graph, it can be shown that the role $\mathtt{r}$ cannot hold [Sir06, Section 5.4.1].

Sirin also proposes to make use of the *query reordering* optimization [Sir06, Section 6.4]. Here, the aim is to prune as early as possible in the atom iteration. Take, for example, $\varphi(x_0, x_1) = \mathtt{C}(x_0) \wedge \mathtt{r}(x_0, x_1) \wedge \mathtt{D}(x_1)$. Let us assume that $|\mathsf{Ind}(\mathtt{C}, \mathcal{K})| = 1\,000$, $|\mathsf{Ind}(\mathtt{D}, \mathcal{K})| = 4$, and $\mathtt{r}$ connects ten individuals in $\mathtt{C}$ to any of the four individuals in $\mathtt{D}$. If we iterate through $\varphi(x_0, x_1)$ in the given order, we first do one instance retrieval on $\mathsf{Ind}(\mathtt{C}, \mathcal{K})$,

which yields 1 000 candidates $\mathtt{i}$ to retrieve $\mathtt{r}$-successors for, i.e., 1 000 instance retrieval calls of the form $\mathsf{Ind}(\exists \mathtt{r}^{-}.\{\mathtt{i}\}, \mathcal{K})$. These retrievals are, however, often unnecessary, as only 1% of them is a member of $\mathtt{D}$. If we rather started with retrieving all $\mathtt{i} \in \mathsf{Ind}(\mathtt{D}, \mathcal{K})$ first and then fetching $\mathsf{Ind}(\exists \mathtt{r}.\{\mathtt{i}\}, \mathcal{K})$, only one instance retrieval for $\mathtt{D}$ and four instance retrievals for $\mathtt{r}$ would have been necessary. Therefore, query order matters drastically in practical settings, but requires determining a good order without knowing both the time required for retrieving $\mathsf{Ind}(\cdot)$ and the size of the returned set beforehand. The latter can be estimated by examining a sample of all individuals to find guarantees for (non-)membership to the concept or role in the completion graph, and extrapolating a size from that. The first can be estimated by checking for how many individuals a guarantee can be given by the completion graph alone (if this ratio is high, the computational effort for $\mathsf{Ind}(\cdot)$ is likely to be low). Moreover, some general, heuristic principles of query reordering can be applied, e.g., to examine roles first as the number of connections between individuals is often low.

All these techniques are predestined when we are confronted with a query that contains at least one answer and no existentially quantified variable, which means we do not need to roll up the query and can examine it atom-by-atom. However, three other cases need to be considered, which can also be practically optimized according to Sirin, Křemen, and Kouba [Sir06; KK11]:

1. We have no answer variable. In this case, the Boolean result is generated by a single consistency check against the rolled-up query.

2. We have exactly one answer variable and at least one quantified variable. In this case, a set of candidates for the variable is generated with the above techniques, the query is rolled up to the answer variable, and the candidates are checked using binary instance retrieval against the rolled-up concept.

3. We have more than one answer variable and at least one quantified variable (the most general case). Here, we can separate the query into multiple parts: one part that has no quantified variables at all, and multiple parts that contain connected, quantified variables (so-called "cores" [KK11]). For the first part, the optimized atom-by-atom method can be applied, pruning large parts of the search space already. By this strategy, query reordering becomes applicable again. Finally, the connected parts with quantified variables are evaluated on the pruned search space. First, a set of candidates for each of the answer variables is generated by treating all other variables as quantified and rolling up the query to the remaining answer variable. This set is intersected with the candidates generated for the first, non-quantified query part. This yields a space of possible answers, which can be evaluated by checking the Boolean query as in case 1.

In total, all four cases yield an optimized CQ answering framework, which is implemented in software tools like Pellet [Sir+07].

3.3 Temporal Querying: State-of-the-Art

The main motivation of this dissertation is to query scenarios, which means that we are inherently confronted with time traces. For a nontemporal query setting, the previously introduced standard CQ answering over expressive DLs is already a good fit, but does not provide means for accessing *discrete temporal data*. Naturally, extensions of queries with temporal operators over KBs have been proposed starting from the early 2000s [Art+02], with software prototypes being presented as early as 2009 [Bar+09].

3.3.1 Temporal Querying in Expressive Description Logics

For temporally querying expressive DLs, an important line of work has theoretically examined answering a temporal extension of conjunctive queries – essentially infinite-time LTL with past operators and CQs – over $\mathcal{ALC}$ to $\mathcal{SHOIQ}$ KBs [BBL13; Lip14; BBL15b; BBL15a]. They have given complexity analyses showing that answering temporal queries over these KBs is ExpTime-complete. In a nutshell, the provided decision procedure is a reduction to non-temporal query answering. The idea is to compile the temporal aspects of the query into an FA whose transitions are checked for satisfiability by answering UCQs. Moreover, the authors allow a part of the roles and concepts to optionally be defined as rigid, i.e., not changing over time. Allowing rigidity increases complexity in some DLs (such as from ExpTime-complete to 2ExpTime-complete for $\mathcal{ALC}$ [BBL13]) and hence it is not considered further in this work. The language has overall been examined from a complexity-theoretic point of view and comparatively hard to implement, as it is defined over infinite traces and allows for mixed past- and future-time operators. As of yet, no query language appears to combine practical implementability with the required expressiveness.

Directly related work on temporally querying expressive DLs – apart from the work of Baader et al. [BBL13; Lip14; BBL15b; BBL15a] as well as the works published in the course of this dissertation [Wes+24a; Wes+24b; WJN25] – is sparse. A proposal by Gutiérrez-Basulto and Klarman offers a broader view on temporal queries over DLs [GK12]. Specifically, they have considered a combination of CQs and the first-order monadic language of order (which subsumes many linear temporal logics). In this modular formalism, called $\mathcal{TQL}$, they have not fixed the DL and put only mild assumptions on the shape of the temporal logic, and have hence obtained complexity results dependent on the choice of formalisms. They have observed that this can lead to Boolean combinations of CQs (BCQs) (queries of the form $\Phi ::= \varphi \mid \neg\varphi \mid \Phi \wedge \Phi$ for CQs φ), where entailment might be intractable or even undecidable in very expressive DLs. Note that Baader et al. have

subsequently shown that BCQ entailment is decidable for $\mathcal{SHIQ}$, $\mathcal{SHOQ}$, and $\mathcal{SHOI}$ by reduction to UCQ entailment [BBL15a], which can be extended to $\mathcal{SROIQ}$ if only simple roles occur in the query [RG10]. As to ease handling BCQs, Gutiérrez-Basulto and Klarman have introduced the autoepistemic **K** operator, which essentially allows a closed-world examination of negated CQs, which in turn paves the way for a straightforward reduction to CQ answering. Using their epistemic semantics, it is easy to see that $\mathcal{TQL}$ inherits complexity from CQ answering if it is PSpace-hard in the given DL, as is, e.g., the case for $\mathcal{ALC}$ (ExpTime-complete). While the assumption of (partially) relying on a CWA might be justified in certain practical settings, this dissertation views the option as a last resort, especially since it counteracts the built-in OWA of DLs. First, other techniques for achieving tractability in answering temporal queries over expressive DLs should be investigated – such as those presented in Chapter 4. If further improvements are required, one can still fall back to a partial CWA. In Section 5.4, we will see why a broad OWA can be desirable.

3.3.2 Temporal Querying in Lightweight Description Logics

A second line of related work examines the case of restricted expressiveness in the DL. As argued in Chapter 2, our setting often requires a higher expressiveness. For completeness, we present the most important related works for the lightweight DLs DL-Lite and $\mathcal{EL}$, which are fragments of $\mathcal{ALC}$. The advantage of lightweight DLs is that the temporal queries and ontology can often be rewritten into standard query languages (without ontologies) [BLT15; Art+22; Art+13], hence allowing to use (temporal) databases.

DL-Lite is a family of DLs optimized for querying large amounts of data and is implemented in the OWL2 QL profile [Art+09]. DL-Lite$_{\text{core}}$ restricts heavily the shape of concept inclusions: the left-hand side allows only for concepts of the form $\bot$, $\mathtt{A}$, $\exists \mathtt{r}.\top$, or $\exists \mathtt{r}^{-}.\top$, and the right-hand side additionally allows for negation. The advantage of DL-Lite is that core reasoning tasks are polynomial in the size of the KB [Cal+07]. More specifically, CQs over DL-Lite$_{\text{core}}$ KBs can be answered in AC^0 – a complexity class contained in P – w.r.t. data complexity. Algorithms for temporally querying DL-Lite KBs have been examined by Borgwardt et al. [BLT13], where, among others, an approach is presented where the temporal query and ontology are rewritten into a temporal database query.

$\mathcal{EL}$ is another lightweight DL with the purpose of allowing efficient, i.e., polynomial time, reasoning in large ontologies [BBL05]. It allows for conjunctions and existential quantification and is the foundation for the OWL2 EL profile. Similar to DL-Lite, temporal queries have been studied by Borgwardt and Thost [BT15], with the result that entailment is polynomial w.r.t. data complexity (when considering neither rigid concepts nor roles). As satisfiability of the underlying temporal formalism (LTL) is PSpace-complete, querying in these lightweight DLs is often also PSpace-complete w.r.t. combined complexity – the

task of querying the KB becomes insignificant.

On the practical side of querying lightweight DLs, reports on first implementations were published. For example, Thost et al. have described QuAnTOn [THÖ15], an implementation of the aforementioned algorithms for temporal queries over DL-Lite [BLT13]. It relies on rewriting into standard SQL, as the authors found the temporal extensions of SQL (at that time) to be insufficiently supported by existing systems. Ontop-temporal extends the existing Ontop reasoner for temporal data and a temporal query language adjacent to SPARQL and SWRL with the usual future and past-time temporal operators [Kal+18]. Similar to QuAnTOn, the input is rewritten into an SQL query and executed on the temporal database.

3.3.3 Temporal Querying in Logic Programming

Combinations of temporal logics with knowledge representation formalisms based on logic programming have also been pursued. A prominent example is Datalog, a subset of the logic programming language Prolog. For example, the OWL2 RL profile can be translated to Datalog. Datalog allows representing facts (data based on predicates) and rules (knowledge in form of definite Horn clauses over the predicates, which is a disjunction with exactly one positive literal). Reasoning in Datalog is P-complete w.r.t. data complexity and ExpTime-complete for combined complexity [Mai+18]. Research on querying temporal logic programming databases based on Datalog has started around 1990 with Datalog_{1S} [Cho90], where Datalog predicates are extended by a special argument for temporal information. More recently, integrations of MTL with Datalog (DatalogMTL) have been examined [Bra+17], also over discrete time [Wał+20] (which subsumes Datalog_{1S}). Answering DatalogMTL queries is ExpSpace-complete; this drops to PSpace-completeness when considering only data complexity. However, for this setting, the software tool MeTeoR provides an efficient implementation also for recursive DatalogMTL based on the standard procedure of materialization (which computes consequences of Datalog programs) in combination with an automaton-based procedure [Wan+22].

Instead of relying on Datalog, it is also possible to rely directly on Prolog, as, e.g., ETALIS does for stream reasoning [Ani+12]. In contrast to other works, ETALIS does not use LTL or MTL but instead leverages Allen's interval logic for temporal queries [All83], which can, for discrete time, be represented by equivalent LTL formulae [RB06].

3.3.4 Temporalization of Knowledge Bases

Besides temporal querying, a plethora of DLs with directly integrated temporal formalisms have been introduced [AF00; LWZ08; Art+14], also on finite traces [AMO18]. In general, two options are possible: to add temporal operators around concept inclusions or inside of DL axioms. An example of the first route is $\mathcal{ALC}$-LTL, where concept inclusions act in place of atomic propositions inside LTL formulae [BGL12], e.g., $\Diamond\Box(\mathtt{A} \sqsubseteq \mathtt{B})$ (at some

point, B will always subsume A). In fact, this formalism can be seen as a special case of the temporal query formalism proposed by Baader et al. [BBL13], as concept inclusions can be modeled using a set of special axioms in the ontology together with a modified temporal query. In $\text{LTL}_{\mathcal{ALC}}$, on the other hand, temporal constructors can be used alongside the standard $\mathcal{ALC}$ operators [LWZ08]. An example is the formula $\texttt{A} \sqsubseteq \Box\texttt{A}$ (membership to A cannot be revoked over time). Moreover, extensions including metric operators have been proposed, such as $\text{MTL}_{\mathcal{ALC}}$ [GJO16; Baa+20], as well as considerations for lightweight DLs [GJK16]. These classical combinations were not conceived in a query answering context, so more recently, several frameworks for addressing that have been introduced [Art+17].

3.4 Metric Temporal Conjunctive Queries

This section introduces the employed temporal formalisms as well as MTCQs, which is the core temporal query language of this dissertation and based on the above introduced CQs. The core idea is to use an extension of LTL_f with metric operators in combination with CQs in place of atomic propositions, which are interpreted over a temporalized DL KB.

The next subsection gives all formal details, i.e., syntax and semantics. For an intuitive overview, we start by referencing back to the requirements derived in Section 2.3. Recall that the language shall

1. be based on a logic that models temporal data with finite duration and finitely many time points,
2. make use of expressive background knowledge,
3. assume incomplete specification and data as well as monotonicity of the inferences,
4. rely on future-time and metric operators, and
5. allow querying the data.

The above sketched idea of using a metric version of LTL_f over CQs defined on temporalized DL KBs satisfies these constraints. For this, the subsequently delineated language

1. combines DLs with a temporal formalism, where the temporal data are represented as a finite sequence of nontemporal DL data,
2. allows relying on expressive DL fragments, as long as no nominals are used and UCQ entailment is decidable,
3. retains the usual DL semantics under which the OWA and monotonicity holds,

4. leverages LTL_f, which operates solely over future-time operators, and adds a metric version of each temporal operator, and

5. uses the set of certain answers semantics, modeling a behavior akin to searching in classical databases.

3.4.1 Syntax and Semantics of Metric Temporal Conjunctive Queries

We first require extending the definition of classical KBs, as given above in Definition 3.2.6, to a temporal one. Many temporal extensions of KBs exists, e.g., allowing to use temporal operators in concepts. Based on the requirements derived in Section 2.3, we opt for a standard extension based on the literature [BBL13], where we keep the modeling capabilities of classical ontologies but extend the data to be temporal. For this, Baader et al. model a TKB by an ontology $\mathcal{O}$ and a finite sequence of assertions that describe the data over time.

Definition 3.4.1 (Temporal Knowledge Base [BBL13])

A Temporal Knowledge Base of length $n \in \mathbb{N}$ is a tuple $\mathcal{K} = (\mathcal{O}, (\mathcal{D}_i)_{i \in \{0,\dots,n\}})$ where $\mathcal{O}$ is an ontology, and each $\mathcal{D}_i$ is a database over the same set of individuals, i.e., $\mathsf{Ind}(\mathcal{D}_0) = \dots = \mathsf{Ind}(\mathcal{D}_n) =: \mathsf{Ind}((\mathcal{D}_i)_{i \in \{0,\dots,n\}})$.

Note that we have not fixed an ontology language, but can, for now, assume to have at least the expressivity of $\mathcal{ALC}$. For brevity, we sometimes denote the i-th non-temporal KB of a TKB by $\mathcal{K}_i = (\mathcal{O}, \mathcal{D}_i)$. The semantics of TKBs is defined by temporal interpretations using the non-temporal case as its basis.

Definition 3.4.2 (Temporal Interpretation [BBL13])

A temporal interpretation $\mathfrak{I}$ is a finite sequence $\mathfrak{I} = (\mathcal{I}_i)_{i \in \{0,\dots,m\}}$ of interpretations over a fixed domain Δ, i.e., $\mathcal{I}_i = (\Delta, \cdot^{\mathcal{I}_i})$, and $m \in \mathbb{N}$ s.t. $a^{\mathcal{I}_i} = a^{\mathcal{I}_j}$, for all $a \in \mathsf{N_I}$ and $0 \leq i, j \leq m$.

Here, we make the *constant domain assumption*, i.e., all interpretations share a common domain. The semantics of TKBs is now defined in terms of temporal interpretations.

Definition 3.4.3 (Semantics of Temporal Knowledge Bases [BBL13])

A temporal interpretation $\mathfrak{I}$ is a model of a Temporal Knowledge Base $\mathcal{K}$ with $\mathcal{K} = (\mathcal{O}, (\mathcal{D}_i)_{i \in \{0,\dots,n\}})$, written $\mathfrak{I} \models \mathcal{K}$, if $m = n$ and for all $i \in \{0, \dots, n\}$ it holds that $\mathcal{I}_i \models \mathcal{K}_i$.

Example 3.4.1

Consider a temporal extension of the KB $\mathcal{K}_h$ introduced in Example 3.2.4, i.e., $\mathcal{K}_t = (\mathcal{O}_h, \mathcal{D}_t)$ with

$$D_t = (\{\texttt{Human}(\texttt{Jon}), \texttt{Human}(\texttt{Jane})\},$$
$$\{\texttt{Human}(\texttt{Jon}), \texttt{Human}(\texttt{Jane}), \texttt{next\_to}(\texttt{Jon}, \texttt{Jane}), \texttt{next\_to}(\texttt{Jane}, \texttt{Jon})\}).$$

At both time points, `Jon` and `Jane` are humans, however, they are only next to each other in the last step. Note that we simply kept the original (non-temporal) ontology. Therefore, both `Jon` and `Jane` must be pedestrians in the second time point in any model $\mathfrak{I}$ of $\mathcal{K}_t$. Formally, for any $\mathfrak{I}$ with $\mathfrak{I} \models \mathcal{K}_t$ it holds that any $a \in \Delta^{\mathcal{I}_1}$ with $a \in \texttt{Jon}^{\mathcal{I}_1}$ or $a \in \texttt{Jane}^{\mathcal{I}_1}$ must satisfy $a \in \texttt{Pedestrian}^{\mathcal{I}_1}$. For $\mathcal{I}_0$, only $a \in \texttt{Human}^{\mathcal{I}_0}$ is guaranteed for $a \in \texttt{Jon}^{\mathcal{I}_0}$ and $a \in \texttt{Jane}^{\mathcal{I}_0}$.

We continue with defining the language of MTCQs that we use to query TKBs. It is a combination of standard CQs with temporal operators inspired by Mission-Time Linear Temporal Logic (MLTL) [LVR19].

Definition 3.4.4 (Syntax of Metric Temporal Conjunctive Queries [Wes+24a])

Let $\mathsf{N_V}$ be a countably infinite set of variable names. MTCQs adhere to the grammar

$$\Phi ::= \varphi \mid \neg\Phi \mid \Phi \wedge \Phi \mid \Phi \mathcal{U} \Phi \mid \Phi \mathcal{U}_{[a,b]} \Phi$$

with φ being a CQ and $a, b \in \mathbb{N}$.

Here, we allow for the standard Boolean operators as well as an unbounded until ($\mathcal{U}$) and a bounded until ($\mathcal{U}_{[a,b]}$). We define the same sets as already done for CQs, e.g., $\mathsf{Ind}(\Phi)$ (the set of individuals) and $\mathsf{Var}(\Phi)$ (the set of variables). $\mathsf{CQ}(\Phi)$ additionally denotes the set of CQs used in Φ, and $\mathsf{Atoms}(\Phi) = \{\mathsf{Atoms}(\varphi) \mid \varphi \in \mathsf{CQ}(\Phi)\}$. An MTCQ Φ is *Boolean* if $\mathsf{AVar}(\Phi) = \emptyset$. Otherwise, we also write $\Phi(\vec{x})$ to denote a non-Boolean query Φ with its list of answer variables $\vec{x}$. An MTCQ of the form $\Phi ::= \varphi \mid \varphi \vee \Phi$, i.e., a disjunction of CQs, is also called a UCQ.

Boolean CQs form the basis for the semantics of Boolean MTCQs.

Definition 3.4.5 (Semantics of Boolean Metric Temporal Conjunctive Queries [Wes+24a])

Let $\mathfrak{I} = (\mathcal{I}_i)_{i \in \{0,\dots,m\}}$ be a temporal interpretation and $i \in \{0, \dots, m\}$. The semantics of Boolean Metric Temporal Conjunctive Queries is given by structural induction:

- $\mathfrak{I}, i \models \Phi$ iff $\mathcal{I}_i \models \Phi$, if Φ is a Boolean CQ;
- $\mathfrak{I}, i \models \neg\Phi$ iff $\mathfrak{I}, i \not\models \Phi$;
- $\mathfrak{I}, i \models \Phi_1 \wedge \Phi_2$ iff $\mathfrak{I}, i \models \Phi_1$ and $\mathfrak{I}, i \models \Phi_2$;
- $\mathfrak{I}, i \models \Phi_1 \mathcal{U}_{[a,b]} \Phi_2$ iff there is a $k \in [a, b]$ with $i + k \leq m$ s.t. $\mathfrak{I}, i + k \models \Phi_2$ and $\mathfrak{I}, i + j \models \Phi_1$, for all $j \in [a, k)$;
- $\mathfrak{I}, i \models \Phi_1 \mathcal{U} \Phi_2$ iff there is a $k \in [i, m]$ s.t. $\mathfrak{I}, k \models \Phi_2$ and $\mathfrak{I}, j \models \Phi_1$, for all $j \in [i, k)$.

We hence use the expected semantics for CQs and the Boolean operators. Unbounded until requires the second subformula to eventually hold at some point in the future and the first subformula to hold at all preceding time points. Its bounded version demands both constraints to hold in the given interval (which diverges slightly from the semantics of bounded until in MTL where this constraint is only put on the right subformula [Koy90]).

We also write $\mathfrak{I} \models \Phi$ for $\mathfrak{I}, 0 \models \Phi$ and allow the typical abbreviations $\Phi_1 \vee \Phi_2$ for $\neg(\neg\Phi_1 \wedge \neg\Phi_2)$, false for $\exists x.\mathtt{A}(x) \wedge \neg\exists x.\mathtt{A}(x)$ for some $\mathtt{A} \in \mathsf{N_C}$, true for $\neg\mathsf{false}$, $\Diamond_{[a,b]}\Phi$ for $\mathsf{true}\mathcal{U}_{[a,b]}\Phi$, $\Diamond\Phi$ for $\mathsf{true}\mathcal{U}\Phi$, $\Box_{[a,b]}\Phi$ for $\neg\Diamond_{[a,b]}\neg\Phi$, and $\Box\Phi$ for $\neg\Diamond\neg\Phi$. The *release operator* $\Phi_1 \mathcal{R} \Phi_2$ is defined as $\neg(\neg\Phi_1 \mathcal{U} \neg\Phi_2)$, with its bounded version $\Phi_1 \mathcal{R}_{[a,b]} \Phi_2$ denoting $\neg(\neg\Phi_1 \mathcal{U}_{[a,b]} \neg\Phi_2)$. The *strong next operator* is defined as $\bigcirc\Phi \equiv \Diamond_{[1,1]}\Phi$ and its dual *weak next* as $\bullet\Phi \equiv \Box_{[1,1]}\Phi$. To avoid having a special case for last or weak next as temporal operators in proofs, we assume an auxiliary predicate last which is true only at the last time point, e.g., by the CQ $\exists x.\mathtt{Last}(x)$ and adding a fresh individual $\mathtt{l}$ with $\mathtt{Last}(\mathtt{l})$ to the last database.

The unconstrained until operator is an extension to the original semantics of MLTL, which adds expressiveness assuming an upper bound on the trace length is not given a-priori. For example, even the simple property $\Box a$ cannot be expressed in MLTL if we are given traces of arbitrary but finite lengths. In the original work on MLTL, this is avoided by assuming that an upper bound $\hat{t}$ of trace lengths can always be derived at time of specification, e.g., traces are at most one hour long. Then, $\Box a$ can be expressed as $\Box_{[0,\hat{t}]} a$. However, in what follows, we do not make such an assumption. Furthermore, note that the finite trace semantics of MTCQs exhibits some non-obvious behaviors, e.g., $\Diamond\Box\Phi$ is equivalent to $\Box\Diamond\Phi$ [GV13].

3.4.2 Relevant Problems over Metric Temporal Conjunctive Queries

Similar to what is established in the literature for Boolean CQs, the central problem over Boolean MTCQs is entailment.

Definition 3.4.6 (Metric Temporal Conjunctive Query Entailment [Wes+24a])

A Temporal Knowledge Base $\mathcal{K}$ entails a Boolean Metric Temporal Conjunctive Query Φ, written $\mathcal{K} \models \Phi$, if for all temporal interpretations $\mathfrak{I} \models \mathcal{K}$ it holds that $\mathfrak{I}, 0 \models \Phi$.

We are again interested in not only entailment of Boolean MTCQs but also giving answers to non-Boolean MTCQs.

Definition 3.4.7 (Metric Temporal Conjunctive Query Answering [Wes+24a])

For a Temporal Knowledge Base $\mathcal{K} = (\mathcal{O}, (\mathcal{D}_i)_{i\in\{0,\dots,n\}})$ and a Metric Temporal Conjunctive Query $\Phi(\vec{x})$, the Metric Temporal Conjunctive Query answering problem asks to compute the set of certain answers $\mathsf{cert}_{\mathcal{K}}(\Phi(\vec{x})) = \{\vec{a} \in \mathsf{Ind}((\mathcal{D}_i)_{i\in\{0,\dots,n\}})^{|\vec{x}|} \mid \mathcal{K} \models \Phi(\vec{a})\}$, where $\Phi(\vec{a})$ denotes the Boolean Metric Temporal Conjunctive Query where the variables in $\vec{x}$ have been uniformly replaced by the individual names in $\vec{a}$.

Similar to what holds for CQs, above definition implies that a Boolean MTCQ is entailed iff the empty tuple is a certain answer.

Example 3.4.2

We can now define an MTCQ to query the temporal data given in $\mathcal{K}_t$ of Example 3.4.1. Take, for example, the query $\Phi_{ex}(x) = \Box\texttt{Human}(x) \wedge \Diamond\texttt{Pedestrian}(x)$, which asks for all individuals that are always human and at some point a pedestrian. As already alluded to in Example 3.2.13, `Jane` and `Jon` can both inferred to be pedestrians at the second time point, and therefore $\mathsf{cert}_{\mathcal{K}_t}(\Phi(x)) = \{\texttt{Jane}, \texttt{Jon}\}$.

A more sophisticated MTCQ involving existentially quantified variables is

$$\begin{aligned}&\Phi_{\exists}(x, y) = \\ &\Box\Big(\exists r.\ \texttt{Vehicle}(x) \wedge \texttt{2\_Lane\_Road}(r) \wedge \texttt{on}(r, x) \wedge \texttt{Parking\_Vehicle}(?y)\Big) \wedge \\ &\Diamond\Big(\texttt{in\_front\_of}(y, x) \wedge \bigcirc\big(\texttt{is\_to\_the\_side\_of}(y, x)\,\mathcal{U}\ \texttt{is\_behind}(y, x)\big)\Big)\end{aligned}$$

Note that the existentially quantified variables are only valid in the local scope of their CQ and cannot be re-used in other parts of the MTCQ. For example, if we were to replace y with an existentially quantified variable in the first CQ, a different parking vehicle might be referred to in the other CQs. Recall that nesting finite trace linear temporal operators

can be tricky; in fact, had we left out the conjunct $\texttt{in\_front\_of}(y, x)$, the formula would have reduced to $\Box(\exists r.\texttt{Vehicle}(x) \wedge \texttt{2\_Lane\_Road}(r) \wedge \texttt{on}(r, x) \wedge \texttt{Parking\_Vehicle}(?y)) \wedge \Diamond\bigcirc\texttt{is\_behind}(y, x)$. The collapse of $\mathcal{U}$ and the CQ $\texttt{is\_to\_the\_side\_of}(y, x)$ originates from the missing "delimiter" $\texttt{in\_front\_of}(y, x)$ at the point in time where the $\Diamond$-subformula is beginning to become satisfied. Without this delimiter, the $\Diamond$-formula simply checks whether $\texttt{is\_behind}(y, x)$ holds at some point, as the left subformula of $\mathcal{U}$ cannot be distinguished from arbitrary data.

3.4.3 Computational Complexity

This dissertation is mostly concerned with bringing an expressive querying language to practical applicability by means of algorithmic advances. Experience, especially in the area of DLs, shows that the computational complexity class of the underlying problem can be predictive of the empirically observed algorithmic performance in some cases. Thus, before we dive into the algorithmic aspects in Chapter 4, we close the introduction to the formal foundations with a discussion on computational complexity.

From here on, we assume that any MTCQ has been transformed into an equivalent TCQ without metric operators (but with $\bigcirc$) by using the following recursive transformation τ, assuming $a \leq b$ [LVR19]:

$$\begin{aligned}
&\tau(\Phi_1 \,\mathcal{U}_{[a,b]} \Phi_2) = \bigcirc\tau(\Phi_1 \,\mathcal{U}_{[a-1,b]} \Phi_2) && \text{if } a > 0,\\
&\tau(\Phi_1 \,\mathcal{U}_{[0,b]} \Phi_2) = \Phi_2 \vee (\Phi_1 \wedge \bigcirc\tau(\Phi_1 \,\mathcal{U}_{[0,b-1]} \Phi_2)) && \text{if } b > 1,\\
&\tau(\Phi_1 \,\mathcal{U}_{[0,0]} \Phi_2) = \Phi_2.
\end{aligned}$$

Note that this leads to an exponential blow-up of the query (essentially, the integer bounds are now encoded unary). As we will show next, an exponential run time can generally not be avoided and the transformation therefore does not introduce additional computational complexity. For the theoretical considerations in this section as well as Chapter 4, we are hence concerned solely with TCQs and omit metric operators for readability and conciseness, e.g., in proofs and algorithms. The implementations described in Section 5.1 includes metric operators again, as they will be part of the practical evaluation of Chapter 5. Therefore, from Section 5.1 on, we again refer to MTCQs.

Let us focus on the prototypical expressive DL $\mathcal{ALC}$ first. Here, both entailment checking and answering of TCQs is ExpTime-complete, and answering does not introduce additional complexity. Observe that query entailment is a lower bound to TCQ answering: Definition 3.4.7 yields, for a Boolean TCQ Φ, either $\mathsf{cert}_{\mathcal{K}}(\Phi) = \{\emptyset\}$ (in which case $\mathcal{K} \models \Phi$) or $\emptyset$ (in which case $\mathcal{K} \not\models \Phi$). Moreover, an upper bound to TCQ answering is provided by an exponential reduction to TCQ entailment. We can just iterate through

all possible answer candidates $\vec{a}$ (of which we have exponentially many in $|\vec{x}|$) and ask whether $\mathcal{K} \models \Phi(\vec{a})$, thus having exponentially many ExpTime entailment checks, which is again in ExpTime.

We now show ExpTime-completeness of TCQ entailment over $\mathcal{ALC}$-TKBs, which can be done by a reduction to the closely related temporal querying setting by Baader et al. [BBL13]. As introduced in Section 3.3, they use mixed past- and future-time LTL operators over infinite words, and hence allow for the following syntax for their queries (denoted TCQ^{BBL} in what follows): $\Psi ::= \varphi \mid \neg\Psi \mid \Psi \wedge \Psi \mid \Psi \,\mathcal{U}\, \Psi \mid \bullet\Psi \mid \Psi \,\mathcal{S}\, \Psi \mid \bigcirc^{-}\Psi$ for CQs φ. Here, $\mathcal{S}$ (since) and $\bigcirc^{-}$ (previous) are the past-time equivalents of $\mathcal{U}$ and $\bigcirc$.

Their semantics, denoted here by $\models^{\text{BBL}}$, is defined using infinite instead of finite temporal interpretations [BBL13]. Such a sequence of interpretation $(\mathcal{I}_h)_{h\geq 0}$ at time i is a model of Ψ, written $(\mathcal{I}_h)_{h\geq 0}, i \models^{\text{BBL}} \Psi$, if one of the following statements holds. First, for formulae Ψ of the form φ, $\neg\Psi$, $\Psi_1 \wedge \Psi_2$, and $\bullet$, the same rules apply as in Definition 3.4.5. Moreover,

- $(\mathcal{I}_h)_{h\geq 0}, i \models^{\text{BBL}} \Psi_1 \,\mathcal{U}\, \Psi_2$ iff there is a $k \geq i$ with $(\mathcal{I}_h)_{h\geq 0}, k \models^{\text{BBL}} \Psi_2$ and $(\mathcal{I}_h)_{h\geq 0}, j \models^{\text{BBL}} \Psi_1$ for any j with $i \leq j < k$,
- $(\mathcal{I}_h)_{h\geq 0}, i \models^{\text{BBL}} \bigcirc^{-}\Psi'$ iff $i > 0$ and $(\mathcal{I}_h)_{h\geq 0}, i-1 \models^{\text{BBL}} \Psi'$, and
- $(\mathcal{I}_h)_{h\geq 0}, i \models^{\text{BBL}} \Psi_1 \,\mathcal{S}\, \Psi_2$ iff there is a $k \leq i$ with $(\mathcal{I}_h)_{h\geq 0}, k \models^{\text{BBL}} \Psi_2$ and $(\mathcal{I}_h)_{h\geq 0}, j \models^{\text{BBL}} \Psi_1$ for any j with $k < j \leq i$.

We say that a TKB $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ of length $n+1$ entails a TCQ^{BBL} Ψ, written $\mathcal{K} \models^{\text{BBL}} \Psi$, iff all infinite sequences of interpretations $(\mathcal{I}_h)_{h\geq 0}$ with $(\mathcal{I}_h)_{h\in\{0,\dots,n\}} \models \mathcal{K}$ and $(\mathcal{I}_h)_{h>n} \models \mathcal{O}$ also satisfy $(\mathcal{I}_h)_{h\geq 0}, n+1 \models^{\text{BBL}} \Psi$. As Baader et al. have shown that deciding $\mathcal{K} \models^{\text{BBL}} \Psi$ is ExpTime-complete, we can embed TCQs into the queries introduced above to achieve an ExpTime upper bound. Additionally, an ExpTime lower bound is given through the ExpTime-completeness of checking consistency of $\mathcal{ALC}$-TKBs, which can be reduced to TCQ answering. The proof extends the sketch presented by Westhofen et al. [Wes+24a].

Theorem 3.4.1

For a Boolean TCQ Φ and an $\mathcal{ALC}$-TKB $\mathcal{K}$, deciding whether $\mathcal{K} \models \Phi$ is ExpTime-complete.

Proof 3.4.1 (of Theorem 3.4.1)

Let $\mathcal{K} = (\mathcal{O}, (\mathcal{D}_i)_{i\in\{0,\dots,n\}})$ be an $\mathcal{ALC}$-TKB and Φ a Boolean TCQ.

Deciding $\mathcal{K} \models \Phi$ can be reduced to checking $\widehat{\mathcal{K}} \models^{\text{BBL}} \widehat{\Phi}$, where $\widehat{\mathcal{K}} = (\mathcal{O}, (\mathcal{D}_{n-i})_{i\in\{0,\dots,n\}})$, and for any TCQ Φ, $\widehat{\Phi}$ replaces any occurrence of $\mathcal{U}$ by $\mathcal{S}$ and $\bigcirc$ by $\bigcirc^{-}$. We now show that $\mathcal{K} \models \Phi$ iff $\widehat{\mathcal{K}} \models^{\text{BBL}} \widehat{\Phi}$. Since $\widehat{\Phi}$ does not contain future time operators, this amounts

to showing that for any model $(\mathcal{I}_i)_{i\in\{0,\dots,n\}}$ of $\mathcal{K}$ it holds that $(\mathcal{I}_i)_{i\in\{0,\dots,n\}}, 0 \models \Phi$ iff for any infinite sequence $(\mathcal{J}_j)_{j\geq 0}$ with $(\mathcal{J}_j)_{j\in\{0,\dots,n\}} \models \widehat{\mathcal{K}}$ it holds that $(\mathcal{J}_j)_{j\geq 0}, n+1 \models^{\mathrm{BBL}} \widehat{\Phi}$. As $(\mathcal{J}_j)_{j\in\{0,\dots,n\}} \models \widehat{\mathcal{K}}$ is equivalent to $(\mathcal{I}_{n-i})_{i\in\{0,\dots,n\}} \models \widehat{\mathcal{K}}$ by definition of $\widehat{\mathcal{K}}$ and there are no future time operators in $\widehat{\Phi}$, it remains to show that $(\mathcal{I}_i)_{i\in\{0,\dots,n\}} \models \Phi$ iff $(\mathcal{I}_{n-i})_{i\in\{0,\dots,n\}} \models \widehat{\Phi}$. The operators $\neg$ and $\wedge$ are not affected by the order of interpretations. The remaining cases are $(\mathcal{I}_i)_{i\in\{0,\dots,n\}}, i \models \bigcirc\Phi'$ iff $(\mathcal{I}_i)_{i\in\{0,\dots,n\}}, n-i \models \bigcirc^{-}\widehat{\Phi}'$ and $(\mathcal{I}_i)_{i\in\{0,\dots,n\}}, i \models \Phi_1 \,\mathcal{U}\, \Phi_2$ iff $(\mathcal{I}_i)_{i\in\{0,\dots,n\}}, n-i \models \widehat{\Phi}_1 \,\mathcal{S}\, \widehat{\Phi}_2$, for which the statement holds by definition of the operators $\bigcirc$ and $\bigcirc^{-}$ as well as $\mathcal{S}$ and $\mathcal{U}$. Thus, checking $\models$ is at most as complex as it is for $\models^{\mathrm{BBL}}$, which is ExpTime-complete for $\mathcal{ALC}$ [BBL13].

ExpTime is also a lower bound for TCQ entailment over $\mathcal{ALC}$, as checking consistency of a KB $\mathcal{K}$ can be reduced to deciding $\mathcal{K} \models \mathsf{false}$ (which is the case iff $\mathcal{K}$ is inconsistent). Therefore, deciding TCQ entailment is ExpTime-complete. □

In general, complexity can shift upwards if we consider more expressive DLs than $\mathcal{ALC}$, such as $\mathcal{ALCI}$ and $\mathcal{SHIQ}$, where query entailment is 2ExpTime-complete [Lut07; Gli+08]. As we will see with both approaches presented in the next sections, TCQ entailment can be decided by exponentially many calls to a CQ engine. Thus, TCQ entailment in a given DL fragment inherits the complexity of CQ entailment, given that the latter is ExpTime-hard.

4 Tractable Temporal Querying over Expressive Description Logics

Contents

We now move towards algorithms for computing the set of certain answers for a given TCQ and TKB with an ontology formulated in an expressive DL. For motivation, consider a naïve approach to computing $\mathsf{cert}_{\mathcal{K}}(\Phi)$: One might assume that it is sufficient to answer all $\mathsf{CQ}(\Phi)$ at time i and simply combine the answers inductively according to the semantics of Φ. However, this is not possible due to the interplay of disjunctions in our query language and expressiveness of the ontology language. An example is the TCQ $\Phi_\vee(x) \coloneqq \mathtt{B}(x) \vee \mathtt{C}(x)$ over the TKB $\mathcal{K}_\vee \coloneqq (\mathtt{A} \sqsubseteq \mathtt{B} \sqcup \mathtt{C}, (\{\mathtt{A}(\mathtt{a})\}))$, where $\mathsf{cert}_{\mathcal{K}_\vee}(\Phi_\vee) = \{(\mathtt{a})\}$. A separate check of $\mathtt{B}(x)$ and $\mathtt{C}(x)$ returns no answer, and inductive combination falsely yields no answer as well. This issue explains the restriction to conjunctions in existing

CQ answering implementations over expressive DLs, as complexity is reduced and various optimizations can be employed (cf. Section 3.2.2). Therefore, and in contrast to CQs or $\mathrm{LTL_f}$ over propositional atoms, we require more involved procedures for checking TCQs.

Two novel procedures tackling the problem are proposed in Section 4.1 and Section 4.2; one based on a translation of the query into finite automata and one relying on rewriting into a normal form. The first approach was presented by Westhofen et al. in 2024 [Wes+24a], the latter was published in 2025 [WJN25]. Both publications are frontiers on the field of temporally querying expressive DLs, where, to the date of writing, only theoretical considerations have been made [BBL13; BBL15b]. The question of the fundamental differences between the frameworks arises naturally; therefore, both algorithms are compared in Section 4.3. In a nutshell, relying on deterministic FA leads to unnecessary checks of UCQs. The discussion extends what is presented by Westhofen et al. [WJN25] by a formal proof that, in fact, the determinization procedure is at fault – however, from a practical perspective, deterministic FAs are hardly avoidable. For this proof, we define $\mathrm{TCQ}_{\mathrm{CQ}^*}$, a syntactic fragment of TCQ in which only CQs can be used for answering. We show that within this fragment rewriting does not use UCQs, whereas the automaton-based approach uses UCQs that could have been avoided under a nondeterministic FA.

Both frameworks operate on arbitrary expressive DL fragments, as long as UCQ entailment is decidable and nominals are not allowed. We require the latter as nominals can force interpretation domains to be of finite but different cardinality. In that case, we cannot guarantee independence of query entailment checks at different time points which renders using established non-temporal query answering procedures difficult. We hence do not fix any DL fragment in what follows but allow for an expressiveness between $\mathcal{ALC}$ and $\mathcal{SRIQ}^{(\mathcal{D})}$. The latter is equivalent to the above introduced $\mathcal{SROIQ}^{(\mathcal{D})}$ but misses the possibility to use nominals in ontology axioms.

4.1 An Automaton-Based Approach

We start with an approach that uses FAs for answering TCQs. The essential idea is to compile the TCQ into an FA and perform a reachability check of its final states on the given temporal data. Formal details are presented later, but for an intuitive understanding, consider the example Boolean TCQ $(\mathtt{B(i)} \wedge \bigcirc \mathtt{D(i)}) \vee \mathtt{C(i)}$ over the TKB $(\{\mathtt{A} \sqsubseteq \mathtt{B} \sqcup \mathtt{C}\}, (\{\mathtt{A(i)}\}), \{\mathtt{D(i)}\})$, which is entailed. Figure 4.1 shows an FA that matches the temporal structure of the TCQ, where the single CQs are interpreted as atomic propositions on the edges, e.g., $\mathtt{B(i)}$ is depicted as the proposition $p_{\mathtt{B(i)}}$.

The next step is to traverse the FA under the given data using a suitable acceptance condition. However, this has to be done with care: the usual way of traversing FAs is not directly applicable here as we are combining the standard semantics of FA with UCQ

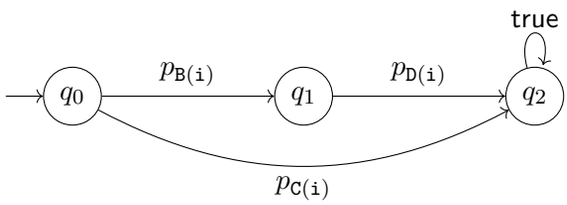

Figure 4.1: The FA for $(\mathtt{B(i)} \wedge \bigcirc \mathtt{D(i)}) \vee \mathtt{C(i)}$, without sink and accepting states.

entailment in expressive DLs. Specifically, when computing successor states during a reachability check, one cannot simply check query entailment for each FA edge. The reason is that usual FAs model disjunctions as transitions into different states (if the two disjuncts require a different behavior of the FA later on). However, as motivated in the introduction to Chapter 4, entailment in expressive DLs does generally not allow the separate examination of disjuncts. For illustration, consider again the example from above. Standard FA semantics demands checking the edges from q_0 to q_1, which is not entailed, and from q_0 to q_2, which is again not entailed. Thus, the query might falsely be rejected as not entailed.

A possible solution is to instead check satisfiability when computing the successor states at a given time point. What is different when considering satisfiability? Here, inversely to entailment, we require evaluating conjunctions together, but can check disjunctions separately. This fits the disjunctive shape of FA naturally, where conjunctions are represented by edge labels and disjunctions are modeled using different successor states. The following presents a framework for TCQ answering based on FAs that uses satisfiability checks and is closely aligned with the standard approach to FA traversal.

4.1.1 Formal Framework

The formal framework is related to the work of Baader et al. [BBL13], whose results we already leveraged for showing ExpTime-completeness of TCQ entailment in Section 3.4.3. The authors made first theoretical considerations on automaton-based temporal query answering over expressive DLs, and, among others, establish an ExpTime upper bound in their setting. For this, they give an algorithm that decides temporal query entailment for the case of infinite trace semantics with mixed past- and future time operators. Due to its theoretical nature, the algorithm is not suitable for practical application. Specifically, for the finite history of the given data of length $n+1$, the approach explicitly checks for each answer candidate and each of the $2^{|\mathsf{CQ}(\Phi)|\cdot(n+1)}$ possible words whether it is valid according to the temporal constraints and satisfies the CQs under the given data. The contribution of this section is to transfer their framework to our setting of finite traces and make it practically applicable. The approach of Baader et al. has two principal steps:

1. it reduces TCQ entailment to satisfiability (for reasons explained above), which then allows to

2. separate the problem into DL and temporal checks, where the latter can be naturally implemented by relying on automata.

For the first step, the essential observation made by Baader et al. is that TCQ entailment is just the inverse of TCQ satisfiability due to the language being closed under negation, i.e.,

$$\begin{aligned}\Phi \text{ satisfiable w.r.t. } \mathcal{K} &\text{ iff there is an } \mathfrak{I} \text{ s.t. } \mathfrak{I} \models \mathcal{K} \text{ and } \mathfrak{I}, 0 \models \Phi \\ &\text{iff not for all } \mathfrak{I} \text{ it holds that not } \mathfrak{I} \models \mathcal{K} \text{ or not } \mathfrak{I}, 0 \models \Phi \\ &\text{iff } \mathcal{K} \not\models \neg\Phi.\end{aligned} \tag{4.1}$$

Example 4.1.1

The TCQ $\Phi_{ex}(\texttt{Jon}) = \Box\texttt{Human}(\texttt{Jon}) \wedge \Diamond\texttt{Pedestrian}(\texttt{Jon})$ derived from Example 3.4.2 is entailed by $\mathcal{K}_t$ of Example 3.4.1 as $\neg\Phi_{ex}(\texttt{Jon}) = \Diamond\neg\texttt{Human}(\texttt{Jon}) \vee \Box\neg\texttt{Pedestrian}(\texttt{Jon})$ is unsatisfiable w.r.t. $\mathcal{K}_t$.

The second step of Baader et al. is to split this satisfiability check into separate DL and temporal components: one where we must check query satisfiability at a given time point and one where we lift these results to the temporal level (for which we can leverage FA). We now adapt this principle to our setting, but introduce two important changes: While Baader et al. use several copies of Büchi automata to decide the temporal problem, we can (more efficiently) rely on a single FA. For the DL component, instead of guessing a sequence of satisfied nontemporal queries, of which there are exponentially many, we traverse the automaton to constructively identify valid sequences.

Let us first introduce some preliminary notation. The propositional abstraction $\mathsf{PA}(\Phi)$ of Φ is the replacement of each $\varphi \in \mathsf{CQ}(\Phi)$ with a propositional variable p_φ. Note that the propositional abstraction of a TCQ is an $\mathrm{LTL_f}$ formula, which is the underlying temporal formalism [GV13] and adheres to the grammar $\psi ::= \mathcal{P} \mid \neg\psi \mid \psi \wedge \psi \mid \psi \mathcal{U} \psi \mid \bigcirc\psi$. This logic is interpreted over finite words of subsets of the propositional variables $\mathcal{P}$, i.e., words of the form $w = w_0 \cdots w_n$ where each $w_i \subseteq \mathcal{P}$ specifies the propositional variables that are satisfied at time point i. It is defined analogously to Definition 3.4.5 but uses propositional variables in place of CQs. An $\mathrm{LTL_f}$ formula ψ is satisfied by some $w \in (2^{\mathcal{P}})^*$ with length $n+1$, written $w \models \psi$, if $w, 0 \models \psi$, where

- $w, i \models A$, for $A \in \mathcal{P}$, iff $A \in w_i$,
- $w, i \models \neg\psi$ iff $w, i \not\models \psi$,

- $w, i \models \psi_1 \wedge \psi_2$ iff $w, i \models \psi_1$ and $w, i \models \psi_2$,
- $w, i \models \psi_1 \mathcal{U} \psi_2$ iff there is a $k \in [i, n]$ s.t. $w, k \models \psi_2$ and $w, j \models \psi_1$ for all $j \in [i, k)$, and
- $w, i \models \bigcirc \psi$ iff $i < n$ and $w, i + 1 \models \psi$.

We define syntactic sugar analogous to Definition 3.4.5, including the operators $\Diamond$ and $\Box$.

Example 4.1.2

The $\mathrm{LTL_f}$ formula $\neg \mathsf{PA}(\Phi_{ex}(\mathtt{Jon})) = \Diamond \neg p_1 \vee \Box \neg p_2$ is a propositional abstraction for the TCQ $\neg \Phi_{ex}(\mathtt{Jon})$ from Example 4.1.1 using the propositions $\mathcal{P} = \{p_1, p_2\}$ as a replacement for the CQs $\mathtt{Human}(\mathtt{Jon})$ and $\mathtt{Pedestrian}(\mathtt{Jon})$. Its models $w \in (2^{\mathcal{P}})^*$ must contain $\emptyset$ or $\{p_2\}$ at least once or $\{p_1\}$ at any point. Note that this includes the empty word ϵ.

We are now ready to transfer the splitting technique proposed by Baader et al. [BBL13, Lemma 3.4] to our problem of deciding TCQ satisfiability, namely by two separate DL and $\mathrm{LTL_f}$ tasks which are connected by the sets of CQs $X_0, \ldots, X_n$.

Lemma 4.1.1 ([Wes+24a])

For a Boolean TCQ Φ and a TKB $\mathcal{K} = (\mathcal{O}, (\mathcal{D}_i)_{i \in \{0,\ldots,n\}})$ over a DL between $\mathcal{ALC}$ and $\mathcal{SRIQ}^{(\mathcal{D})}$, Φ is satisfiable w.r.t. $\mathcal{K}$ iff there is a sequence $X_0, \ldots, X_n$ of subsets of $\mathsf{CQ}(\Phi)$ s.t.:

1. there are interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_n$ s.t. for all $i \in \{0, \ldots, n\}$ we have $\mathcal{I}_i \models \mathcal{K}_i$ and $\mathcal{I}_i \models \varphi$ for every $\varphi \in X_i$, and $\mathcal{I}_i \not\models \varphi$ for every $\varphi \in \mathsf{CQ}(\Phi) \setminus X_i$, and
2. $(\{p_\varphi \mid \varphi \in X_i\})_{i \in \{0,\ldots,n\}} \models \mathsf{PA}(\Phi)$.

Example 4.1.3

For the Boolean TCQ $\neg \Phi_{ex}(\mathtt{Jon})$ from Example 4.1.1 and the TKB $\mathcal{K}_t$ from Example 3.4.1, there can be no sequence X_0, X_1 of CQs that satisfies the constraints of Lemma 4.1.1. By satisfying $\neg \mathsf{PA}(\Phi_{ex}(\mathtt{Jon}))$ as per Example 4.1.2, X_0, X_1 must contain $\emptyset$ or $\{\mathtt{Pedestrian}(\mathtt{Jon})\}$ at least once or $\{\mathtt{Human}(\mathtt{Jon})\}$ at any point. Now consider interpretations $\mathcal{I}_0, \mathcal{I}_1$. In case that X_0, X_1 contains $\emptyset$ or $\{\mathtt{Pedestrian}(\mathtt{Jon})\}$, Point 2 requires for some $i \in \{0, 1\}$ that $\mathcal{I}_i \not\models \mathtt{Human}(\mathtt{Jon})$ holds, which cannot be satisfied due to the assertions in the data of $\mathcal{K}_t$. If, on the other hand, $\{\mathtt{Human}(\mathtt{Jon})\}$ always holds, Point 2 demands that $\mathcal{I}_i \not\models \mathtt{Pedestrian}(\mathtt{Jon})$ for all $i \in \{0, 1\}$. This can again not be satisfied: For the second data point, as described in Example 3.2.13, $\mathtt{Jon}$ is inferred to always be a $\mathtt{Pedestrian}$. Hence, $\neg \mathsf{PA}(\Phi_{ex}(\mathtt{Jon}))$ is unsatisfiable w.r.t. $\mathcal{K}_t$ and thus, by Equation (4.1), $\mathcal{K}_t \models \Phi_{ex}(\mathtt{Jon})$.

The following proof of Lemma 4.1.1 extends the sketch originally given by Westhofen et al. [Wes+24a].

Proof 4.1.1 (of Lemma 4.1.1)

For the proof, we first show that we can safely assume that $\mathcal{I}_0, \dots, \mathcal{I}_n$ are over the same domain. More specifically, we can combine $\mathcal{I}_0, \dots, \mathcal{I}_n$ witnessing Point 1 in Lemma 4.1.1 but with potentially different domains into $\mathcal{I}'_0, \dots, \mathcal{I}'_n$ with the same domain using a standard argument, cf. the proof of Theorem 5.21 by Lippmann [Lip14]. Since DLs between $\mathcal{ALC}$ and $\mathcal{SRIQ}^{(\mathcal{D})}$ (without the universal role) cannot enforce finite models [Rud11], we can assume that each $\mathcal{I}_i$ is infinite. By the downward Löwenheim-Skolem theorem, we can assume that the $\mathcal{I}_i$ are countably infinite and thus have the same domain. Had we allowed nominals, the ontology can control the finiteness of their domains, e.g., by using the axiom $\top \sqsubseteq \{\mathtt{a}\}$ for some individual $\mathtt{a}$, and hence the assumption that all interpretations are over the same domain breaks. For example, the TKB $(\{\top \sqsubseteq \{\mathtt{a}, \mathtt{b}\}, \mathtt{A} \sqsubseteq \{\mathtt{a}\}\}, (\emptyset, \{\mathtt{A}(\mathtt{a}), \mathtt{A}(\mathtt{b})\}))$ has only models where the first interpretation has a cardinality of at most two, i.e., it is allowed that $a^{\mathcal{I}_0} \neq b^{\mathcal{I}_0}$, whereas the domain of the second interpretation must be a singleton, i.e., $a^{\mathcal{I}_1} = b^{\mathcal{I}_1}$. For constructing $\mathcal{I}'_0, \dots, \mathcal{I}'_n$, the final step is to identify all individual names in the domains.

The remainder of the proof is similar to what has been shown by Lippmann for the case of infinite words [Lip14, Lemma 5.18]. For this, we define the propositional abstraction of a temporal interpretation $\mathfrak{I}$ of length m as $\mathsf{PA}(\mathfrak{I}) = (\{p_\varphi \mid \varphi \in \mathsf{CQ}(\Phi),\ \mathcal{I}_i \models \varphi\})_{i \in \{0,\dots,m\}}$. From the similarities in the semantics of $\mathrm{LTL_f}$ and TCQs it follows that

$$\mathfrak{I}, i \models \Phi \text{ iff } \mathsf{PA}(\mathfrak{I}), i \models \mathsf{PA}(\Phi) \tag{4.2}$$

for any $i \in \{0, \dots, m\}$ and TCQ Φ, which can be shown by induction: For a CQ ψ, $\mathsf{PA}(\psi) = p_\psi$, and hence $\mathcal{I}, i \models \psi$ iff $p_\psi \in \{p_\varphi \mid \varphi \in \mathsf{CQ}(\Phi),\ \mathcal{I}_i \models \varphi\} = \mathsf{PA}(\mathfrak{I})_i$ iff $\mathsf{PA}(\mathfrak{I}), i \models \mathsf{PA}(\psi)$. For the induction step, consider a TCQ Φ. For TCQs Φ_1 and Φ_2, the followings cases must be considered:

- $\mathfrak{I}, i \models \neg\Phi_1$ iff $\mathfrak{I}, i \not\models \Phi_1$ and, by the induction hypothesis, $\mathsf{PA}(\mathfrak{I}), i \not\models \mathsf{PA}(\Phi_1)$ iff $\mathsf{PA}(\mathfrak{I}), i \models \mathsf{PA}(\neg\Phi_1)$,
- $\mathfrak{I}, i \models \Phi_1 \vee \Phi_2$ iff $\mathfrak{I}, i \models \Phi_1$ or $\mathfrak{I}, i \models \Phi_2$ and, by the induction hypothesis, $\mathsf{PA}(\mathfrak{I}), i \models \mathsf{PA}(\Phi_1)$ or $\mathsf{PA}(\mathfrak{I}), i \models \mathsf{PA}(\Phi_2)$ iff $\mathsf{PA}(\mathfrak{I}), i \models \mathsf{PA}(\Phi_1 \vee \Phi_2)$,
- $\mathfrak{I}, i \models \bigcirc\Phi_1$ iff $i < m$ and $\mathfrak{I}, i+1 \models \Phi_1$ and, by the induction hypothesis, $\mathsf{PA}(\mathfrak{I}), i+1 \models \mathsf{PA}(\Phi_1)$ iff $i < m$ and $\mathsf{PA}(\mathfrak{I}), i+1 \models \mathsf{PA}(\Phi_1)$ iff $\mathsf{PA}(\mathfrak{I}), i \models \mathsf{PA}(\bigcirc\Phi_1)$, and
- $\mathfrak{I}, i \models \Phi_1 \mathcal{U} \Phi_2$ iff there is a $k \in [i, m]$ s.t. $\mathfrak{I}, k \models \Phi_2$ and $\mathfrak{I}, j \models \Phi_1$ for all $j \in [i, k)$ and, by the induction hypothesis, $\mathsf{PA}(\mathfrak{I}), k \models \mathsf{PA}(\Phi_2)$ and $\mathsf{PA}(\mathfrak{I}), j \models \mathsf{PA}(\Phi_1)$ iff there is a $k \in [i, m]$ s.t. $\mathsf{PA}(\mathfrak{I}), k \models \mathsf{PA}(\Phi_2)$ and $\mathsf{PA}(\mathfrak{I}), j \models \Phi_1$, for all $j \in [i, k)$ iff $\mathsf{PA}(\mathfrak{I}), i \models \mathsf{PA}(\Phi_1 \mathcal{U} \Phi_2)$.

From Equation (4.2), it follows that Points 1 and 2 of Lemma 4.1.1 imply satisfiability of Φ w.r.t. $\mathcal{K}$: Consider $\mathfrak{I} = (\mathcal{I}_i)_{i\in\{0,\dots,n\}}$ with $\mathcal{I}_i$ given by Point 1, which asserts $\mathfrak{I}$ to be a model of $\mathcal{K}$. For this, the introductory assumption of each $\mathcal{I}_i$ having the same domain must hold. By Point 2, $\mathsf{PA}(\mathfrak{I}) \models \mathsf{PA}(\Phi)$. Equation (4.2) then yields $\mathfrak{I} \models \Phi$.

It remains to show the other direction, i.e., satisfiability of Φ w.r.t. $\mathcal{K}$ implies existence of the mentioned sequence $X_0, \dots, X_n$ of subsets of $\mathsf{CQ}(\Phi)$. Let $\mathfrak{I}$ be a temporal interpretation with $\mathfrak{I} \models \mathcal{K}$ and $\mathfrak{I} \models \Phi$, and $i \in \{0, \dots, n\}$. Consider now $X_i = \{\varphi \in \mathsf{CQ}(\Phi) \mid \mathcal{I}_i \models \varphi\}$. Point 1 is satisfied by the non-temporal interpretations contained in $\mathfrak{I}$: $\mathcal{I}_i \models \mathcal{K}_i$ holds by the assumption that $\mathfrak{I}$ is a model of $\mathcal{K}$, and $\mathcal{I}_i \models \varphi$ for every $\varphi \in X_i$ holds by construction of X_i. Additionally, $\mathcal{I}_i \not\models \varphi$ for $\varphi \notin X_i$ is given, as otherwise, i.e., in case $\mathcal{I}_i \models \varphi$, φ would have been in X_i. Point 2 follows from the construction of X_i and Equation (4.2). □

We now show how we implement the check of Point 1 of Lemma 4.1.1, similarly to what has been done by Lippmann [Lip14, Theorem 5.10]. We first remark that Lemma 4.1.1 does not enforce a connection between the interpretations $\mathcal{I}_0, \dots, \mathcal{I}_n$, meaning that the checks at different time points are independent of each other. By the natural connection between satisfiability and entailment, we can leverage an engine for answering disjunctions of CQs over non-temporal $\mathcal{ALC}$ KBs checking Point 1, i.e., computing $\mathsf{cert}_{\mathcal{K}}(\Phi)$ for $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ and Φ a disjunction of CQs. For doing so, we associate with every Boolean CQ φ its canonical database $\mathcal{D}_\varphi$, which is the set of all conjuncts that occur in φ, i.e., $\mathcal{D}_\varphi \coloneqq \{v(\alpha) \mid \alpha \in \mathsf{Atoms}(\varphi)\}$, where $v(\alpha)$ replaces in α any quantified variable name of φ with a unique, fresh individual name w.r.t. $\mathcal{K}$. Then, Point 1 of Lemma 4.1.1 can be reduced to UCQ checks in conjunction with canonical databases. Note that we do not make the UNA – Lippmann additionally shows how to incorporate the UNA in such a setting, if desired [Lip14, Theorem 5.10].

Lemma 4.1.2 ([Wes+24a])

Let X be a set of Boolean CQs and $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ a KB over a DL between $\mathcal{ALC}$ and $\mathcal{SRIQ}^{(\mathcal{D})}$. The following two statements are equivalent for every subset $Z \subseteq X$:

(a) There is a model $\mathcal{I}$ of $\mathcal{K}$ s.t. $\mathcal{I} \models \varphi$ for every $\varphi \in Z$, and $\mathcal{I} \not\models \varphi$ for every $\varphi \in X \setminus Z$.

(b) $(\mathcal{O}, \mathcal{D}') \not\models \bigvee_{\varphi\in X\setminus Z} \varphi$ where $\mathcal{D}'$ is the union of $\mathcal{D}$ with $\mathcal{D}_\varphi$ for each $\varphi \in Z$ (with variables across different $\mathcal{D}_\varphi$ suitably renamed).

Example 4.1.4

Assume the example set of CQs $X = \{\texttt{Human}(\texttt{Jon}), \texttt{Pedestrian}(\texttt{Jon})\}$. If we desire checking satisfiability of the overall TCQ $\neg\Phi_{ex}(\texttt{Jon})$ w.r.t. $(\mathcal{O}_h, \mathcal{D}_h)$ from Example 3.2.4 via Lemma 4.1.1, we might encounter the subset $X_0 = \{\texttt{Human}(\texttt{Jon})\} \subset X$. We can

now use Lemma 4.1.2 to check Point 2 of Lemma 4.1.1 by assessing whether $(\mathcal{O}_h, \mathcal{D}_h \cup \{\texttt{Human}(\texttt{Jon})\}) \not\models \texttt{Pedestrian}(\texttt{Jon})$. The query is, however, entailed, as `Jon` must be a pedestrian in $(\mathcal{O}_h, \mathcal{D}_h)$, and hence X_0 does not satisfy Point 2 of Lemma 4.1.1.

Proof 4.1.2 (of Lemma 4.1.2)

Let X be a set of Boolean CQs, $Z \subseteq X$, and $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ a KB as mentioned in the Lemma. We first show that Point (b) is equivalent to there being an interpretation $\mathcal{J}$ with (i) $\mathcal{J} \models \mathcal{K}$, (ii) $\mathcal{J} \not\models \varphi$ for all $\varphi \in X \setminus Z$, and (iii) $\mathcal{J} \models \mathcal{D}_\varphi$ for all $\varphi \in Z$: By definition of entailment, Point (b) holds iff there is no interpretation $\mathcal{J}$ for which $(\mathcal{O}, \mathcal{D}') \not\models \mathcal{J}$ or $\mathcal{J} \models \varphi$ for some $\varphi \in X \setminus Z$, which in turn holds iff there is an interpretation $\mathcal{J}$ with $\mathcal{J} \models (\mathcal{O}, \mathcal{D}')$ and $\mathcal{J} \not\models \varphi$ for all $\varphi \in X \setminus Z$. Additionally, $\mathcal{J} \models (\mathcal{O}, \mathcal{D}')$ iff $\mathcal{J} \models \mathcal{K}$ and $\mathcal{J} \models \mathcal{D}_\varphi$ for all $\varphi \in Z$.

The proof of Lemma 4.1.2 is split in two directions. Let us first assume Point (b), i.e., the existence of an interpretation $\mathcal{J}$ with the mentioned properties (i) to (iii). Point (a) requires, besides properties (i) and (ii), also $\mathcal{J} \models \varphi$ for every $\varphi \in Z$. By definition, $\mathcal{J} \models \mathcal{D}_\varphi$ iff for all atoms $\alpha \in \mathsf{Atoms}(\varphi)$ it holds that $\mathcal{J} \models v(\alpha)$ where v replaces any quantified variable with a fresh individual name. Consider $\pi : \mathsf{Var}(\varphi) \cup \mathsf{Ind}(\varphi) \to \Delta^{\mathcal{J}}$ with $\pi(\mathtt{a}) = \mathtt{a}^{\mathcal{J}}$ for any individual $\mathtt{a}$ and $\pi(x) = (v(x))^{\mathcal{J}}$ for any variable x. As $\mathcal{J} \models \mathcal{D}_\varphi$, it holds that $\pi(x) \in \mathtt{C}^{\mathcal{J}}$ for all $\mathtt{C}(x) \in \mathsf{Atoms}(\varphi)$ and $(\pi(x), \pi(x)') \in \mathtt{r}^{\mathcal{J}}$ for all $\mathtt{r}(x, x') \in \mathsf{Atoms}(\varphi)$, which is equivalent to $\mathcal{J} \models \varphi$. Thus, $\mathcal{J}$ satisfies all requirements of Point (a).

For the other direction, assume Point (a), i.e., there is an interpretation $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$, $\mathcal{I} \models \varphi$ for every $\varphi \in Z$, and $\mathcal{I} \not\models \varphi$ for every $\varphi \in X \setminus Z$. Proving Point (b) amounts to showing properties (i) to (iii) hold, where (i) and (ii) are trivially satisfied by any interpretation that extends $\mathcal{I}$ by fresh individual names only. For property (iii), we hence define $\mathcal{J}$ as an extension of $\mathcal{I}$. Recall that $\mathcal{I} \models \varphi$ iff $\pi : \mathsf{Var}(\varphi) \cup \mathsf{Ind}(\varphi) \to \Delta^{\mathcal{I}}$ with $\pi(\mathtt{a}) = \mathtt{a}^{\mathcal{I}}$ for all individuals $\mathtt{a}$, $\pi(x) \in \mathtt{C}^{\mathcal{I}}$ for all $\mathtt{C}(x) \in \mathsf{Atoms}(\varphi)$, and $(\pi(x), \pi(x)') \in \mathtt{r}^{\mathcal{I}}$ for all $\mathtt{r}(x, x') \in \mathsf{Atoms}(\varphi)$. We define $\mathcal{J} = (\Delta^{\mathcal{I}} \cup \{v(x) \mid x \in \mathsf{Var}(\varphi), \varphi \in Z\}, \cdot^{\mathcal{I}} \cup \{(v(x), \pi(x)) \mid x \in \mathsf{Var}(\varphi), \varphi \in Z\})$. Let $\varphi \in Z$ and $\alpha \in \mathsf{Atoms}(\varphi)$. If $\alpha = \mathtt{C}(x)$, it follows from the assumption that $\pi(x) \in \mathtt{C}^{\mathcal{I}}$ and hence $(v(x))^{\mathcal{J}} = \pi(x) \in \mathtt{C}^{\mathcal{J}}$, i.e., $\mathcal{J} \models \mathtt{C}(v(x))$. The case of $\alpha = \mathtt{r}(x, x')$ is analogous and therefore $\mathcal{J}$ is a model of all axioms in the canonical database of φ, i.e., $\mathcal{J} \models \mathcal{D}_\varphi$, implying Point (b). □

Lemma 4.1.2 hence gives us a way to use UCQ answering algorithms for Point 1 of Lemma 4.1.1.

For Point 2 of Lemma 4.1.1, we can rely on the established means of checking the $\mathrm{LTL_f}$ word problem by translation to an FA and a reachability check w.r.t. its final states. Recall that Point 2 amounts to checking whether the sequence of propositions corresponding to the satisfied queries (a finite word) satisfies the propositional abstraction of the TCQ (an $\mathrm{LTL_f}$ formula). Such an $\mathrm{LTL_f}$ formula ψ over propositional variables $\mathcal{P}$

can be represented by a deterministic or nondeterministic FA A_ψ over the alphabet $2^{\mathcal{P}}$ s.t. for any $\tau \in (2^{\mathcal{P}})^*$ it holds that $\tau \models \psi$ iff $\tau \in L(A_\psi)$ [GV13].

The transformation of LTL_f to nondeterministic FAs is worst-case exponential – any LTL_f formula can be linearly translated into an equivalent Alternating Finite Automaton (AFA), for which a corresponding nondeterministic FA with exponentially many states can be created. For deterministic FA, we contrarily have a worst-case doubly-exponential transformation [GF21]. For example, we can again linearly convert an LTL_f formula to an AFA, which can then be converted to a deterministic FA with worst-case doubly-exponential state space [CKS81]. There is, in general, no efficient conversion of LTL_f formulae to FAs. Despite that, several viable implementations were proposed [Zhu+17; Ban+20; GF21]. All of them have in common that they rely on deterministic instead of nondeterministic FA, as they make heavy use of intermediate minimization steps to achieve adequate performance – or, in some cases, even timely termination [KMS02, Section 4.3]. Since minimization of nondeterministic FA is PSpace-complete [JR93], state-of-the-art algorithms perform a preliminary determinization step prior to minimization. Due to this reason, our practical implementation presented later will rely on deterministic FA as well.

The subsequently presented TCQ answering algorithm, however, assumes the general case of having a nondeterministic FA with a transition relation. This implies that it can also be directly used with deterministic FAs (by simply encoding the transition function as a relation). In the following, we thus refer simply to FAs. To conclude, Point 2 of Lemma 4.1.1 can be checked by means of a reachability check of the final states in the FA A_ψ under the sequence of the CQs $X_0, \ldots, X_n$ identified in Point 1 of Lemma 4.1.1.

Example 4.1.5

Figure 4.2 shows the FA for the LTL_f formula $\mathsf{PA}(\neg\Phi_{ex}(x))$, i.e., $A_{\mathsf{PA}(\neg\Phi_{ex}(x))}$. The transitions are labeled using a symbolic encoding, where Boolean formulae over the propositions indicate a transition for each model of the formula. For example $\neg p_{\texttt{Human}(x)}$ represents the transitions $\emptyset$ and $\texttt{Pedestrian}(x)$. This encoding can be exponentially more succinct than the explicit one assumed by the definitions above and is subsequently used for ease of presentation.

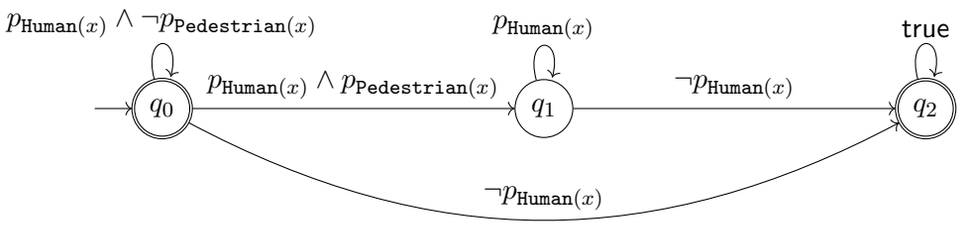

Figure 4.2: An FA representing the temporal structure of $\mathsf{PA}(\neg(\Box\texttt{Human}(x) \wedge \Diamond\texttt{Pedestrian}(x)))$.

The language of $A_{\mathsf{PA}(\neg\Phi_{ex}(x))}$ contains exactly the words described in Example 4.1.2. Note that the given automaton is already deterministic and minimal – there is no deterministic FA with two states accepting $L(A_{\mathsf{PA}(\neg\Phi_{ex}(x))})$.

4.1.2 Automaton-Based Algorithm

We now present an algorithm that computes the set of certain answers by using the introduced indirection through satisfiability. A direct implementation of the adapted framework of Baader et al. (given in Lemma 4.1.1) is, however, highly impractical, as it involves guessing possible sequences of CQs and, for each, checking whether Points 1 and 2 are satisfied. We hence contribute a novel, practical approach to implementing Lemma 4.1.1, namely, by a constructive technique for building the sequence $X_0, \ldots, X_n$ that relies in each step on the already identified subsequence $X_0, \ldots, X_{i-2}$ to find the new X_{i-1}, for $i \in \{1, \ldots, n+1\}$.

Our basic procedure for computing $\mathsf{cert}_{\mathcal{K}}(\Phi(\vec{x}))$ is depicted in Algorithm 1, taken from Westhofen et al. [Wes+24a]. Intuitively, it identifies an abstraction of all possible "runs" $X_0, \ldots, X_n$ in a step-by-step fashion by using FA states and checks Points 1 and 2 of Lemma 4.1.1 in each step. For this, the algorithm first constructs the FA $A_{\mathsf{PA}(\neg\Phi(\vec{x}))}$ for the negated propositional abstraction of $\Phi(\vec{x})$ in order to reduce entailment to satisfiability as described earlier. The core data structure is the function $S : \mathsf{Ind}(\mathcal{K})^{|\vec{x}|} \times \{0, \ldots, n+1\} \to Q$, where $S(\vec{a}, i)$ contains all states of the FA we can reach after i steps for each answer candidate $\vec{a}$. At time $i > 0$, we put a state q in $S(\vec{a}, i)$ iff there is a sequence $X_0, \ldots, X_{i-2}$ that satisfies Point 1 of Lemma 4.1.1 and it can be extended by X_{i-1} s.t. it satisfies Point 2 as well. For this, the algorithm initializes S with q_0 for all candidates $\vec{a}$. It then computes $S(\vec{a}, i)$ by checking for which previously reachable states $q \in S(\vec{a}, i-1)$ – behind which lies a valid sequence $X_0, \ldots, X_{i-2}$ – there is a predicate P for which $X_{i-1} = \{\varphi \mid p_\varphi \in P\}$ completes the sequence. Note that it is not necessary to store the sequences $X_0, \ldots, X_{i-1}$ explicitly, as it suffices to work only on the states that generated the sequences.

The newly identified subset $X_{i-1} \subseteq \mathsf{CQ}(\Phi(\vec{x}))$ has two properties: First, it satisfies Point 1 of Lemma 4.1.1, which is done by the central test $\mathsf{Sat}^{\mathcal{K}_{i-1}}_{\neg\Phi(\vec{a})}(\cdot)$ in Line 7. This

Algorithm 1 Computing certain answers to TCQs [Wes+24a].

Input: TCQ $\Phi(\vec{x})$, TKB $\mathcal{K} = (\mathcal{O}, (\mathcal{D})_{i\in\{0,\dots,n\}})$

Output: $\mathsf{cert}_{\mathcal{K}}(\Phi(\vec{x}))$

1: $(Q, \Sigma, q_0, \Delta, F) \coloneqq A_{\mathsf{PA}(\neg\Phi(\vec{x}))}$
2: Initialize $S(\vec{a}, 0) \coloneqq \{q_0\}$ for all $\vec{a} \in \mathsf{Ind}(\mathcal{D})^{|\vec{x}|}$
3: **for** $i \coloneqq 1$ to $n+1$ **do**
4: **for** $\vec{a} \in \mathsf{Ind}(\mathcal{D})^{|\vec{x}|}$ **do**
5: $S(\vec{a}, i) \coloneqq \emptyset$
6: **for** $q \in S(\vec{a}, i-1)$ **do**
7: $S(\vec{a}, i) \coloneqq S(\vec{a}, i) \cup \{q' \in Q \mid (q, P, q') \in \Delta \text{ and } \mathsf{Sat}^{\mathcal{K}_{i-1}}_{\neg\Phi(\vec{a})}(\{\varphi \mid p_\varphi \in P\})\}$
8: **end for**
9: **end for**
10: **end for**
11: **return** $\{\vec{a} \in \mathsf{Ind}(\mathcal{D})^{|\vec{x}|} \mid S(\vec{a}, n+1) \cap F = \emptyset\}$;

requires checking a condition of shape (a) in Lemma 4.1.2, which we do by using its reformulation as shape (b). This allows us to use a non-temporal query engine for UCQs. Note that this is more involved than answering each disjunction separately, a problem already known to the DL community. For correctly answering UCQs, we require a reformulation of the disjunction into Conjunctive Normal Form (CNF) and then answer each conjunct separately as described by Horrocks et al. [HT00]. For a Boolean TCQ Φ and set $Z \subseteq \mathsf{CQ}(\Phi)$, we define $\mathsf{Sat}^{\mathcal{K}}_{\Phi}(Z)$ as true iff $\mathcal{K}$ and Z pass the test in Point (b) of Lemma 4.1.2 and thus Point 1 of Lemma 4.1.1.

The second required property of X_{i-1} is to allow a completion of $X_0, \dots, X_{i-2}$ into what eventually becomes the sequence $X_0, \dots, X_n$ that satisfies the temporal constraint of Point 2 of Lemma 4.1.1. The algorithm ensures this by taking only valid successor states $(q, P, q') \in \Delta$ in the FA. Note that the algorithm must use a nondeterministic interpretation even when given a deterministic FA, as DLs allow two different subsets of CQs X_k and X_l to be satisfiable under the same data. Therefore, for a given time, data point, and state, multiple states might be reachable even in a deterministic FA; we therefore simply assume a transition relation in Line 1.

Finally, the algorithm returns all $\vec{a}$ for which no final state is reachable after $n+1$ steps. This means that there is a sequence of transitions $(q_0, P_0, q_1) \in \Delta, \dots, (q_n, P_n, q_{n+1}) \in \Delta$ with $q_{n+1} \notin F$ for which the induced sequence of CQ subsets $\{\varphi \mid p_\varphi \in P_0\}, \dots, \{\varphi \mid p_\varphi \in P_{n+1}\}$ satisfies Points 1 and 2 Lemma 4.1.1. It therefore follows that the algorithm accepts exactly those candidates $\vec{a}$ not satisfying the negated query and thus $\vec{a} \in \mathsf{cert}_{\mathcal{K}}(\Phi)$.

We can apply some standard improvements on Algorithm 1. For example, one can work directly on a minimal FA or remove $\vec{a}$ from the candidates if $S(\vec{a}, i)$ contains an accepting sink, i.e., an accepting state for which no other state is reachable, which means

that $\vec{a}$ must always satisfy the negated TCQ. Additionally, in practice, we of course rely on symbolically instead of explicitly encoded transitions.

Example 4.1.6

Let us again consider the example TCQ $\Phi_{ex}(x)$, for which the FA of its negated propositional abstraction is given in Figure 4.2. The algorithm identifies a candidate (a, b) as a certain answer iff the FA ends up in state q_1 in all possible runs, according to the TKB. Recall that multiple transitions might be satisfiable at any given time, e.g., the empty KB satisfies the transitions $\neg p_{\texttt{Human}(x)}$ and $p_{\texttt{Human}(x)} \wedge \neg p_{\texttt{Pedestrian}(x)}$ of state q_0. Therefore, it is not sufficient to search for a run of the FA that ends up in q_1; rather, *all* runs have to finish there. The only way to achieve this is for the FA to not stay in q_0 or q_2. For this, it has to eventually change from q_0 to q_1 by having neither $\texttt{Human}(a) \wedge \neg\texttt{Pedestrian}(a)$ nor $\neg\texttt{Human}(a)$ but $\texttt{Human}(a) \wedge \texttt{Pedestrian}(a)$ satisfiable. The FA shall then stay solely in q_1 with only $\texttt{Human}(a)$ satisfiable for the remainder. It can then be concluded that (a, b) must be a certain answer to $\Phi_{ex}(x)$.

The correctness of the overall formal approach to answering TCQs via FAs has been already shown in Lemma 4.1.1. It remains to show that Algorithm 1 does, in fact, implement the framework sketched in Lemma 4.1.1, for which correctness has already been shown in Proof 4.1.1.

Theorem 4.1.1 (Correctness of Algorithm 1)

For a TCQ $\Phi(\vec{x})$ and TKB $\mathcal{K} = (\mathcal{O}, (\mathcal{D}_i)_{i\in\{0,\dots,n\}})$ over a DL between $\mathcal{ALC}$ and $\mathcal{SRIQ}^{(\mathcal{D})}$, Algorithm 1 returns $\mathsf{cert}_{\mathcal{K}}(\Phi(\vec{x}))$.

Proof 4.1.3 (of Theorem 4.1.1)

Let $\Phi(\vec{x})$ be a TCQ and $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ a TKB. For a candidate $\vec{a} \in \mathsf{Ind}(\mathcal{D})^{|\vec{x}|}$, $\vec{a} \in \mathsf{cert}_{\mathcal{K}}(\Phi(\vec{x}))$ is equivalent to $\neg\Phi(\vec{a})$ not being satisfiable w.r.t. $\mathcal{K}$, which again is equivalent to $A_{\mathsf{PA}(\neg\Phi(\vec{a}))}$ having no path $q_0 \overset{P_0}{\to} q_1 \dots q_n \overset{P_n}{\to} q_{n+1}$ with $q_{n+1} \in F$ and, for any $i \in \{0, \dots, n\}$, $\mathsf{Sat}^{\mathcal{K}_i}_{\neg\Phi(\vec{a})}(\{\varphi \in \mathsf{CQ}(\Phi(\vec{a})) \mid p_\varphi \in P_i\})\}$. The first equivalence is asserted by Equation (4.1). The second equivalence holds as Points 1 and 2 of Lemma 4.1.1 are respected: Since any P_i on the path satisfies $\mathsf{Sat}^{\mathcal{K}_i}_{\neg\Phi(\vec{a})}(\{\varphi \in \mathsf{CQ}(\Phi(\vec{a})) \mid p_\varphi \in P_i\})\}$ by assumption and Sat implements the check defined in Point (b) of Lemma 4.1.2, the sequence $X_0, \dots, X_n$ with $X_i = \{\varphi \in \mathsf{CQ}(\Phi(\vec{a})) \mid p_\varphi \in P_i\}$ satisfies Point 1 of Lemma 4.1.1. Moreover, Point 2 of Lemma 4.1.1 is directly implied by the correctness of $A_{\mathsf{PA}(\neg\Phi)}$ [GV13], i.e., for any path in $q_0 \overset{P_0}{\to} q_1 \dots q_n \overset{P_n}{\to} q_{n+1}$ with $q_{n+1} \in F$, $P_0 \dots P_n \models \neg\mathsf{PA}(\Phi(\vec{a}))$.

It hence remains to show that Algorithm 1 checks wether no such path $q_0 \overset{P_0}{\to} q_1 \dots q_n \overset{P_n}{\to} q_{n+1}$ with the constraints mentioned above exists. For convenience, we assume that the

iteration over all candidates $\vec{a} \in \mathsf{Ind}(\mathcal{D})^{|\vec{x}|}$ (Line 4) is given as an outermost loop and hence the algorithm examines a Boolean query $\Phi(\vec{a})$. Consider the sequence of states $q_0 \in S(\vec{a}, 0), \ldots, q_{n+1} \in S(\vec{a}, n+1)$. For any pair q_i, q_{i+1}, $i \leq n$, it holds that there exists some P_i s.t. $(q_i, P_i, q_{i+1}) \in \Delta$ by the correct selection of q_i in Lines 3 and 6 and the check of the transition relation Δ in Line 7. The call to Sat in Line 7 ensures that any such P_i satisfies $\mathsf{Sat}^{\mathcal{K}_i}_{\neg\Phi(\vec{a})}(\{\varphi \in \mathsf{CQ}(\Phi(\vec{a})) \mid p_\varphi \in P_i\})\}$. Finally, the acceptance condition in Line 11 corresponds to rejecting any path which does not adhere to the above construction or for which $q_{n+1} \in F$. Hence, Algorithm 1 checks whether no path through the FA with the mentioned constrains exists. The preliminary equivalences hence assert that Algorithm 1 returns exactly any candidate for which $\vec{a} \in \mathsf{cert}_{\mathcal{K}}(\Phi(\vec{x}))$ holds. □

4.1.3 Computational Complexity

We conclude the discussion of Algorithm 1 by showing that it uses exponential time over TKBs with $\mathcal{ALC}$-ontologies when used with nondeterministic FAs. As TCQ entailment is ExpTime-complete for $\mathcal{ALC}$, Algorithm 1 can offer optimal run time. Optimality here refers to the fact that the run time of Algorithm 1 matches the proven lower bound for the complexity of the underlying entailment problem. In practice, however, deterministic FAs are often a more feasible choice (for reasons discussed above), and therefore, software implementations might exhibit a doubly-exponential dependency on the $\mathrm{LTL_f}$ formula due to its conversion into a deterministic FA. In theory, this can be avoided with nondeterministic FAs, which is hence what the following discussion assumes.

Theorem 4.1.2

For a TCQ $\Phi(\vec{x})$ and an $\mathcal{ALC}$-TKB $\mathcal{K} = (\mathcal{O}, (\mathcal{D}_i)_{i \in \{0,\ldots,n\}})$, Algorithm 1 runs in exponential time.

Proof 4.1.4 (of Theorem 4.1.2)

Algorithm 1 first constructs $A_{\mathsf{PA}(\neg\Phi(\vec{x}))}$, which can be done in exponential time if the automaton is nondeterministic [GV13]. Subsequently, given the assumed explicit transition encoding, it executes the automaton under the data, thereby checking, for each of the $n+1$ time points and $|\mathsf{Ind}(\mathcal{D})^{|\vec{x}|}|$ many individuals, a transition from any state in Q to some other state in Q labeled with some subset of propositions. Note that the set of reachable states is bounded by the automaton's state space and hence at most exponential in the size of the query: $|S(\vec{a}, i)| \leq |Q| \in O(2^{|\neg\Phi(\vec{x})|})$ for any $i \leq n+1$. Formally, this leads to, at most,

$$(n+1) \cdot |\mathsf{Ind}(\mathcal{D})^{|\vec{x}|}| \cdot |Q|^2 \cdot 2^{|\mathsf{CQ}(\Phi(\vec{x}))|}$$

calls to $\mathsf{Sat}^{\mathcal{K}_{i-1}}_{\neg\Phi(\vec{a})}$.

We can show that $\mathsf{Sat}^{\mathcal{K}_{i-1}}_{\neg\Phi(\vec{a})}(\{\varphi \mid p_\varphi \in P\})$ for a given transition $(q, P, q') \in \Delta$ and $i \in \{1, \dots, n+1\}$ has an exponential run time: Recall that this function tests whether $(\mathcal{O}, \mathcal{D})$ and $Z = \{\varphi \in \mathsf{CQ}(\Phi(\vec{a})) \mid p_\varphi \in P\}$ pass Point (b) of Lemma 4.1.2, i.e., checking whether $(\mathcal{O}, \mathcal{D}') \not\models \bigvee_{\varphi \in \mathsf{CQ}(\Phi(\vec{a}))\setminus Z} \varphi$, where $\mathcal{D}'$ is the union of $\mathcal{D}$ with the canonical database $\mathcal{D}_\varphi$ for each $\varphi \in Z$ (with variables across different $\mathcal{D}_\varphi$ suitably renamed). Note that $|\mathcal{D}'| \leq |\mathcal{D}| + |\mathsf{CQ}(\Phi(\vec{a}))| \cdot |\mathsf{Atoms}(\Phi(\vec{a}))|$ and $|\bigvee_{\varphi \in \mathsf{CQ}(\Phi(\vec{a}))\setminus Z} \varphi| \leq |\mathsf{CQ}(\Phi(\vec{a}))| \cdot |\mathsf{Atoms}(\Phi(\vec{a}))|$, i.e., the input to the UCQ entailment problem is polynomial w.r.t. $\mathcal{K}$ and $\Phi(\vec{a})$. For $\mathcal{ALC}$, UCQ entailment is ExpTime-complete [Ort10, Theorem 6.5.1], and hence $\mathsf{Sat}^{\mathcal{K}_{i-1}}_{\neg\Phi(\vec{a})}(\{\varphi \mid p_\varphi \in P\})$ runs in exponential time as well. □

Note that UCQ entailment can become 2ExpTime-complete for more complex DLs, such as $\mathcal{ALCHI}$ [Ort10, Theorem 6.5.2] or $\mathcal{SHIQ}$ [Gli+08, Corollary 33]. The computational complexity of Algorithm 1 is therefore influenced by the underlying DL.

4.1.4 Leveraging Conjunctive Query Answering

In the above specified version, Algorithm 1 has practical applicability, but still leaves open the question of how to efficiently handle the many answer candidates. Of these, in practice, only few are entailed. Any of the candidates is considered individually by Algorithm 1, for which then a similar task (checking query satisfiability) is performed. To avoid this, we can leverage existing systems that implement efficient algorithms specifically tailored towards answering CQs over standard (non-temporal) KBs in order to avoid expensive UCQ checks. They are often able to operate on large sets of candidates instead of considering each candidate separately by using the techniques described in Section 3.2.2, such as binary instance retrieval [SP06].

As a motivation, consider again the FA in Figure 4.2, where $q_2 \in S(\vec{a}, i)$ for all $\vec{a}$, i for which $\neg\texttt{Human}(\vec{a})$ is satisfiable w.r.t. $(\mathcal{O}, \mathcal{D}_{i-1})$. This is, however, equivalent to checking $(\mathcal{O}, \mathcal{D}_{i-1}) \not\models \texttt{Human}(\vec{a})$. As q_2 is an accepting sink, we can hence simply identify the certain answers to the CQ $\texttt{Human}(x)$ using the efficient systems described in Section 3.2.2 and then instantly reject all candidates that are not within the set of certain answers. In practice, many candidates will satisfy this requirement, as entailment requires strong axiomatization of the ontology or data. This allows us to (ideally) filter out many candidates early on and therefore greatly reduce the run time of Algorithm 1.

The concept of extracting certain (non-)answers by answering the CQs occurring in the edges has been generalized by Westhofen et al. [Wes+24a]. We directly see that some satisfiability checks of Algorithm 1 can be fully replaced by a call to a CQ engine, e.g., checking satisfiability of $\neg\texttt{Human}(x)$ as above. This is obviously not possible for all checks. But even for more complicated queries, such as $\neg\texttt{Human}(x) \wedge \texttt{Pedestrian}(x)$, it is possible to use CQ answering to extract information on satisfiability. Let us assume a succinctly encoded edge $\alpha = \bigwedge_{p_\varphi \in P_0} \neg p_\varphi \wedge \bigwedge_{p_\varphi \in P_1} p_\varphi$ for $P_0, P_1 \subseteq P$ (as in Figure 4.2). At a given

time i, a CQ engine can then compute $\mathsf{cert}_{\mathcal{K}_i}(\varphi)$ for all $\varphi \in \{\psi \mid p_\psi \in P_0\} \cup \{\bigwedge_{p_\psi \in P_1} \psi\}$. This allows us to conclude the following on satisfiability:

1. for all $\vec{a} \notin \mathsf{cert}_{\mathcal{K}_i}(\varphi)$: $\neg\varphi(\vec{a})$ is *satisfiable* w.r.t. $\mathcal{K}_i$ (by duality of non-entailment and satisfiability);
2. for all $\vec{a} \in \mathsf{cert}_{\mathcal{K}_i}(\varphi)$: $\varphi(\vec{a})$ is *satisfiable* (as entailment implies satisfiability) and and $\neg\varphi(\vec{a})$ is *unsatisfiable* w.r.t. $\mathcal{K}_i$ (again by duality).

Subsequently, we can transfer this newly gained satisfiability information to the satisfiability of the overall edge α as per Table 4.1. Similar to the function $\mathsf{Sat}^{\mathcal{K}_{i-1}}_{\neg\Phi(\vec{a})}$ for checking

Table 4.1: Transferability of the satisfiability information of a candidate $\vec{a}$ and CQ φ to the satisfiability of $\vec{a}$ for an edge α containing φ. ✓ indicates satisfiability, x unsatisfiability, and ? no transfer being possible.

Shape of edge α	$\varphi(\vec{a})$ unsatisfiable	$\varphi(\vec{a})$ satisfiable
$\alpha = \varphi$	✗	✓
$\alpha = \neg\varphi$	✓	✗
$\varphi \in \alpha$	✗	?
$\neg\varphi \in \alpha$	✓	?

satisfiability of an explicitly encoded edge, we call the function that returns the values described in Table 4.1 $\mathsf{SatApprox}^{\mathcal{K}_{i-1}}_{\neg\Phi(\vec{a})}(\alpha, \vec{a})$ for a symbolically encoded edge α and an answer candidate $\vec{a}$.

It remains to incorporate $\mathsf{SatApprox}^{\mathcal{K}_{i-1}}_{\neg\Phi(\vec{a})}(\alpha, \vec{a})$ into Algorithm 1. For this, the main idea is to perform a prior, under-approximating traversal of the FA. Whereas Algorithm 1 uses the central data structure $S(\vec{a}, i)$ to store the set of reachable states for candidate $\vec{a}$ after i steps, we now use sets $R(\vec{a}, i) \subseteq S(\vec{a}, i)$ and $U(\vec{a}, i) \subseteq Q \setminus S(\vec{a}, i)$ to under-approximate the reachable and unreachable states, respectively.

Algorithm 2 shows how the sets R and U are constructed prior to running Algorithm 1. While the principle for constructing R based on satisfiability knowledge in Line 9 is similar to how Algorithm 1 builds S in Line 7, unsatisfiability knowledge requires adaptations. Firstly, we can only add q' to $U(\vec{a}, i)$ if *all* states q with edges to q' also agree on unsatisfiability of $\vec{a}$ at time i (Lines 16 to 20). Secondly, unsatisfiability generates new satisfiability information: for a reachable state and candidate we know that if all but one of its successor states are unreachable, the single state on which no information is present must be reachable (Lines 22 to 26).

When implementing Algorithm 2, further optimizations can be added: Sinks, i.e., states where the only successor state is the state itself, must not be checked; (un)reachability

Algorithm 2 Computing the reachable and unreachable states R and U.

Input: TCQ, TKB $\mathcal{K} = (\mathcal{O}, (\mathcal{D})_{i\in\{0,\dots,n\}})$
Output: functions $R, U : \mathsf{Ind}(\mathcal{K})^{|\vec{x}|} \times \{0, \dots, n\} \to Q$

$(Q, \Sigma, q_0, \Delta, F) \coloneqq A_{\mathsf{PA}(\neg\Phi)}$
$C \coloneqq \mathsf{Ind}(\mathcal{D})^{|\vec{x}|}$
Initialize $R(\vec{a}, 0) \coloneqq \{q_0\}$ and $U(\vec{a}, 0) \coloneqq Q \setminus \{q_0\}$ for all $\vec{a} \in C$
for $i \coloneqq 1$ to $n + 1$ **do**
  for $\vec{a} \in C$ **do**
    ▷ Computes reachable states and unsatisfied transitions
    $R(\vec{a}, i), U'(q, \vec{a}, i) \coloneqq \emptyset$
    for $q \in R(\vec{a}, i - 1)$ **do**
      $R(\vec{a}, i) \coloneqq R(\vec{a}, i) \cup \{q' \in Q \mid (q, \alpha, q') \in \Delta \text{ and } \mathsf{SatApprox}^{\mathcal{K}_{i-1}}_{\neg\Phi(\vec{a})}(\alpha, \vec{a}) = \checkmark\}$
      $U'(q, \vec{a}, i) \coloneqq \{q' \in Q \mid (q, \alpha, q') \in \Delta \text{ and } \mathsf{SatApprox}^{\mathcal{K}_{i-1}}_{\neg\Phi(\vec{a})}(\alpha, \vec{a}) = \text{✗}\}$
    end for
    ▷ Computes unreachable states:
    ▷ a) by being "too far away" and thus trivially unreachable
    $U(\vec{a}, i) \coloneqq \{q \in Q \mid \text{there is no path } q_0 \to^j q \text{ with } j \leq i\}$
    ▷ b) by unreachability based on unsatisfiability agreement of all valid transitions
    for $q' \in Q$ **do**
      if $q' \in U'(q, \vec{a}, i)$ for all q with $(q, \cdot, q') \in \Delta$ and $q_0 \to^j q$ for some $j < i$ **then**
        $U(\vec{a}, i) \coloneqq U(\vec{a}, i) \cup \{q'\}$
      end if
    end for
    ▷ Bonus: Adds state as reachable if there is a transition from a reachable state for which all other transitions lead to an unreachable state
    for $q \in R(\vec{a}, i - 1)$ **do**
      if there is a q' s.t. $(q, \cdot, q') \in \Delta$ and all $q'' \neq q'$ with $(q, \cdot, q'') \in \Delta$ satisfy $q'' \in U(\vec{a}, i)$ **then**
        $R(\vec{a}, i) \coloneqq R(\vec{a}, i) \cup \{q'\}$
      end if
    end for
  end for
end for
return R, U;

information can simply be copied from the last time point. For clarity and comparability with Algorithm 1, the candidate iteration is made explicit in Algorithm 2. When using a CQ answering engine for implementing $\mathsf{SatApprox}$, the algorithm can be restructured to more efficiently operate on candidate sets instead of using an explicit iteration. While

the above proposal stores (un)satisfiabiltiy information for FA states, practical implementations may opt to also store this information for the constraints on the edges of the FA, further alleviating the burden on the subsequent full traversal.

Algorithm 1 can now be adapted to incorporate the returned reachable and unreachable states R and U from Algorithm 2. First, it can already extract some certain answers from U and some certain non-answers from R, namely the sets $\{\vec{a} \in C \mid U(\vec{a}, n+1) \supseteq F\}$ and $\{\vec{a} \in C \mid R(\vec{a}, n+1) \subseteq F\}$, respectively. These candidates are not considered anymore during the main iteration of Algorithm 1. Second, the algorithm can re-use cached answers to CQs in the first traversal: Instead of initializing $S(\vec{a}, i)$ with $\emptyset$ in Line 5 of Algorithm 1, we set $S(\vec{a}, i)$ to $R(\vec{a}, i)$. Then, in Line 7, we do not require checking reachability of the successor states $q' \in Q$ that are already contained in $S(\vec{a}, i)$ or for which we have unreachability information, i.e., are in $U(\vec{a}, i)$.

Example 4.1.7

Consider again the example Boolean TCQ $\Phi_{ex}(\texttt{Jon})$ over the TKB $\mathcal{K}_t$. The corresponding FA of the negated TCQ is given in Figure 4.2. We now demonstrate that the CQ optimization can avoid some (but not all) UCQs checks. The optimized algorithm first computes the sets $R(\texttt{Jon}, 1) = \emptyset$, $U(\texttt{Jon}, 1) = \{q_2\}$, $R(\texttt{Jon}, 2) = U(\texttt{Jon}, 2) = \emptyset$. We gained the information $U(\texttt{Jon}, 1) = \{q_2\}$ as `Jon` is guaranteed to be a `Human` at any time point and hence cannot satisfy the sink state. The information that q_2 is not reachable in the first iteration is not sufficient to conclude entailment yet, and a full run is required. This full run can now reuse the information that q_2 is unreachable in iteration $i = 1$, and thus a lower number of query entailment checks are required. However, during the full run, we still need to check whether q_0 and q_1 are valid successors to q_0, as acceptance of the given data differs between both states.

4.2 A Rewriting-Based Approach

While the FA-based algorithm has theoretically optimal run time, we cannot expect nondeterministic FAs to be available in software implementations (as discussed above). In practice, we therefore must assume the traversed FA to be deterministic. Alas, even in its optimized variant, Algorithm 1 in combination with deterministic FAs still exhibits inefficient behavior: Despite the aim of avoiding UCQs, it unnecessarily relies on UCQ checks, which are, in practice, more expensive to answer than CQs.

These unnecessary UCQ checks can be exemplified using the above Example 4.1.7. Note that the deterministic FA of Figure 4.2 contains two conjunctions of CQs on the edges outgoing from q_0. Indeed, satisfiability of both conjunctions must be checked on $\mathcal{K}_t$, even when using the CQ optimization. While the algorithm was able to conclude that q_2 is unreachable from q_0 and q_1, a full run was required to assess whether to accept

the given data or not, i.e., whether q_0 or q_1 are reachable from q_0. For this, the full run requires checking the satisfiability of the queries $\texttt{Human}(\texttt{Jon}) \wedge \neg\texttt{Pedestrian}(\texttt{Jon})$ and $\texttt{Human}(\texttt{Jon}) \wedge \texttt{Pedestrian}(\texttt{Jon})$, which both entail UCQ checks. The latter, e.g., is translated per Lemma 4.1.1 to non-entailment of $\neg\texttt{Human}(\texttt{Jon}) \vee \neg\texttt{Pedestrian}(\texttt{Jon})$, which is a UCQ.

Yet, it can be proven that, in fact, no UCQ checking is required for answering the given TCQ and TKB, as the following equivalence holds:

$$\mathcal{K}_t \models \Box\texttt{Human}(\texttt{Jon}) \wedge \Diamond\texttt{Pedestrian}(\texttt{Jon}) \iff$$
$$\mathcal{K}_t \models (\texttt{Human}(\texttt{Jon}) \wedge \bigcirc\texttt{Human}(\texttt{Jon})) \wedge (\texttt{Pedestrian}(\texttt{Jon}) \vee \bigcirc\texttt{Pedestrian}(\texttt{Jon}))$$

Note that the only disjunction within the equivalent formula, namely $\texttt{Pedestrian}(\texttt{Jon}) \vee \bigcirc\texttt{Pedestrian}(\texttt{Jon})$, is not a UCQ as both disjuncts refer to different time points. Their results can hence be safely combined without the interaction between disjuncts that is possible in expressive DLs.

Using the equivalent query from above, a correct procedure for checking $\Box\texttt{Human}(\texttt{Jon}) \wedge \Diamond\texttt{Pedestrian}(\texttt{Jon})$ over $\mathcal{K}_t$ can be devised:

1. Translate $\Box\texttt{Human}(\texttt{Jon}) \wedge \Diamond\texttt{Pedestrian}(\texttt{Jon})$ to $(\texttt{Human}(\texttt{Jon}) \wedge \bullet\texttt{Human}(\texttt{Jon})) \wedge (\texttt{Pedestrian}(\texttt{Jon}) \vee \bigcirc\texttt{Pedestrian}(\texttt{Jon}))$,
2. check if $(\mathcal{K}_t)_0 \models \texttt{Human}(\texttt{Jon})$ (true) and $(\mathcal{K}_t)_0 \models \texttt{Pedestrian}(\texttt{Jon})$ (false),
3. check if $(\mathcal{K}_t)_1 \models \texttt{Human}(\texttt{Jon})$ (true) and $(\mathcal{K}_t)_1 \models \texttt{Pedestrian}(\texttt{Jon})$ (true),
4. substitute these results in $(\texttt{Human}(\texttt{Jon}) \wedge \bullet\texttt{Human}(\texttt{Jon})) \wedge (\texttt{Pedestrian}(\texttt{Jon}) \vee \bigcirc\texttt{Pedestrian}(\texttt{Jon}))$, leading to $(\mathsf{true} \wedge \bullet\mathsf{true}) \wedge (\mathsf{false} \vee \bigcirc\mathsf{true})$, and
5. evaluate the $\mathrm{LTL_f}$ formula; which yields an "entailed" overall.

This shows that $\mathcal{K}_t \models \Box\texttt{Human}(\texttt{Jon}) \wedge \Diamond\texttt{Pedestrian}(\texttt{Jon})$ *can* be decided without answering UCQs. Yet, as demonstrated above, the optimized automaton-based approach of Section 4.1 checks UCQs when traversing the deterministic FA.

4.2.1 Rewriting Algorithm

Practical experience with Algorithm 1 shows that UCQ checks are more expensive than the highly optimized CQ answering procedure presented in Section 3.2.2. This is also the rationale behind the improvement given by Algorithm 2. However, on the example above, even this optimized version used UCQ checks despite them being not required for answering the query. Therefore, a novel algorithm for answering TCQs that uses UCQs more sparingly is in order. For this, we generalize the rewriting approach used in the example above.

Rewriting

As demonstrated in Step 1 of the above example, it is straightforward to translate the temporal operators in the query step-by-step into $\bigcirc$/$\bullet$-sequences. Care has to be taken, however, to ensure that it is possible to combine the answers to subformulae (as done in Step 4 of the example above). The expressiveness of the employed DLs can lead to cases where, e.g., disjuncts interfere with each other, and checking both disjuncts separately leads to incorrect results. As a motivation, take the query $\bigcirc\mathtt{A}(\mathtt{a}) \vee \bigcirc\mathtt{B}(\mathtt{a})$ over the TKB $(\mathtt{C} \sqsubseteq \mathtt{A} \sqcup \mathtt{B}, (\emptyset, \{\mathtt{C}(\mathtt{a})\}))$. Since it contains only $\bigcirc$-operators, it does not require further temporal expansion. Yet, for the second data point, it holds that $(\mathtt{C} \sqsubseteq \mathtt{A} \sqcup \mathtt{B}, \{\mathtt{C}(\mathtt{a})\}) \not\models \mathtt{A}(\mathtt{a})$ and $(\mathtt{C} \sqsubseteq \mathtt{A} \sqcup \mathtt{B}, \{\mathtt{C}(\mathtt{a})\}) \not\models \mathtt{B}(\mathtt{a})$ and hence the overall result would yield "not entailed" as well. This is, however, wrong, as the query is entailed. The correct way is to instead rewrite the TCQ to $\bigcirc(\mathtt{A}(\mathtt{a}) \vee \mathtt{B}(\mathtt{a}))$ first and then check $(\mathtt{C} \sqsubseteq \mathtt{A} \sqcup \mathtt{B}, \{\mathtt{C}(\mathtt{a})\}) \models \mathtt{A}(\mathtt{a}) \vee \mathtt{B}(\mathtt{a})$ for the second data point. In that way, we can safely use inductive combination also for disjunctions involving temporal operators.

Moreover, instead of completely rewriting the TCQ in one pass (as done in Step 1 of the motivating example), we propose a step-by-step method of rewriting to an equivalent TCQ for which the current database can be checked fully, and subsequently advancing to the next time point. For each subformula, the rewriting procedure can be repeated. In this way, no unnecessary rewriting is performed. For example, encountering a conjunct with no certain answers allows discarding the other conjunct completely.

The presented rewriting procedure, called $\mathsf{Transform}(\Phi)$, hence relies on two sets of rules: the standard expansion laws [BK08] and rules to correctly distribute $\bigcirc$ over $\vee$. Through both, we establish the following shape of the TCQ: Temporal formulae are wrapped into $\bigcirc$-formulae and disjunctions have at most one temporal disjunct. This normal form ensures that for any disjunction that is referring to the current time point we can just inductively combine answers to its disjuncts (if it is temporal) or must only answer a query, possibly with disjunctions and negations, over a single database (if it is non-temporal). We call the latter queries Unions of Possibly Negated Conjunctive Queries (UnCQs).

Definition 4.2.1 (Union of Possibly Negated Conjunctive Query)

A Union of Possibly Negated Conjunctive Queries is a Temporal Conjunctive Query of the form $\bigvee_{i=1,\dots,k} \varphi_i \vee \bigvee_{j=k+1,\dots,l} \neg\varphi_j$ for Conjunctive Queries φ_i and φ_j.

As a preliminary step for rewriting, we assume the input Φ to $\mathsf{Transform}$ to be in NNF, where negations appear only in front of CQs. This enables us to consider negations only when checking non-temporal queries. An NNF can be achieved in linear time by exploiting the duality of operators. Due to this, our transformation rules must, besides the primitives $\wedge$, $\bigcirc$, and $\mathcal{U}$, also consider their dual operators $\vee$, $\bullet$, and $\mathcal{R}$.

To achieve the normal form, $\mathsf{Transform}(\Phi)$ applies the rules of Figure 4.3 to any subformula in Φ not occurring within a temporal operator ($\bigcirc$, $\bullet$, $\mathcal{U}$, or $\mathcal{R}$), until no more rule can be fired. Rule 1 distributes next over disjunctions (as motivated earlier), Rule 2 uses distributivity of $\vee$ over $\wedge$ to establish a CNF over non-temporal and next formulae, and Rules 3 and 4 apply the expansion law. Rule 5 allows to disregard $\bullet$ using last, for simplifying further operations. In general and for the transformation rules specifically, we assume commutativity and associativity of $\vee$, i.e., $\Phi_1 \vee \Phi_2 = \Phi_2 \vee \Phi_1$ and $(\Phi_1 \vee \Phi_2) \vee \Phi_3 = \Phi_1 \vee (\Phi_2 \vee \Phi_3)$. For example, Rule 1 on $\bigcirc A(x) \vee B(x) \vee \bigcirc C(x)$ results in $B(x) \vee \bigcirc(A(x) \vee C(x))$.

1. $\bigcirc\Phi_1 \vee \bigcirc\Phi_2 \rightsquigarrow \bigcirc(\Phi_1 \vee \Phi_2)$
2. $\Phi_1 \vee (\Phi_2 \wedge \Phi_3) \rightsquigarrow (\Phi_1 \vee \Phi_2) \wedge (\Phi_1 \vee \Phi_3)$
3. $\Phi_1 \,\mathcal{U}\, \Phi_2 \rightsquigarrow \Phi_2 \vee (\Phi_1 \wedge \bigcirc(\Phi_1 \,\mathcal{U}\, \Phi_2))$
4. $\Phi_1 \,\mathcal{R}\, \Phi_2 \rightsquigarrow \Phi_2 \wedge (\Phi_1 \vee \bullet(\Phi_1 \,\mathcal{R}\, \Phi_2))$
5. $\bullet\Phi \rightsquigarrow \mathsf{last} \vee \bigcirc\Phi$

Figure 4.3: Transformation rules used by $\mathsf{Transform}(\Phi)$ on all subformulae not wrapped in a temporal operator [WJN25].

$\mathsf{Transform}(\Phi)$ has two central properties that are necessary for a correct TCQ answering algorithm. First, Lemma 4.2.1 states that the original semantics are preserved. Second, Lemma 4.2.2 establishes a correct way of inductively computing $\mathsf{cert}_{\mathcal{K}_{\geq j}}(\Phi)$.

Lemma 4.2.1 ([WJN25])

$\mathfrak{I}, 0 \models \Phi$ iff $\mathfrak{I}, 0 \models \mathsf{Transform}(\Phi)$ for any TCQ Φ and temporal interpretation $\mathfrak{I}$.

Lemma 4.2.2 ([WJN25])

Let Φ be a TCQ and $\Phi' := \mathsf{Transform}(\Phi)$. Then Φ' takes the form of a conjunction

$$\textstyle\bigwedge_{k\in\{1,\dots,u\}} \Phi_k \wedge \bigwedge_{l\in\{u+1,\dots,v\}} \Phi_l \wedge \bigwedge_{m\in\{v+1,\dots,w\}} \Phi_m,$$

where each Φ_k is a UnCQ, each $\Phi_l = \bigcirc\Phi'_l$, and each $\Phi_m = \bigcirc\Phi^1_m \vee \Phi^2_m$, with Φ^2_m being a UnCQ. Moreover,

$$\begin{aligned}\mathsf{cert}_{\mathcal{K}_{\geq j}}(\Phi') = \textstyle\bigcap_{k\in\{1,\dots,u\}} \ & \mathsf{cert}_{\mathcal{K}_j}(\Phi_k) \cap \\ \textstyle\bigcap_{l\in\{u+1,\dots,v\}} \ & \mathsf{cert}_{\mathcal{K}_{\geq j+1}}(\Phi'_l) \cap \\ \textstyle\bigcap_{m\in\{v+1,\dots,w\}} & \big(\mathsf{cert}_{\mathcal{K}_{\geq j+1}}(\Phi^1_m) \cup \mathsf{cert}_{\mathcal{K}_j}(\Phi^2_m)\big).\end{aligned}$$

Here, for a TKB $\mathcal{K} = (\mathcal{O}, (\mathcal{D}_{i\in\{0,\dots,n\}}))$ and $j \leq n$, we define $\mathcal{K}_{\geq j} \coloneqq (\mathcal{O}, (\mathcal{D}_{i\in\{j,\dots,n\}}))$. If $j > n$, $\mathcal{K}_{\geq j}$ is undefined, and we use the convention that $\mathsf{cert}_{\mathcal{K}_{\geq j}}(\Phi) = \emptyset$. Recall as well that $\mathcal{K}_j \coloneqq (\mathcal{O}, \mathcal{D}_j)$. We begin with proving Lemma 4.2.1, which is done by showing the statement separately for each rewriting rule of Figure 4.3.

Proof 4.2.1 (of Lemma 4.2.1 [WJN25])

We show that applying any of the five rules does not change semantics for any temporal interpretation $\mathfrak{I} = (\mathcal{I}_i)_{i\in\{0,\dots,n\}}$. Let $i \in \{0, \dots, n\}$.

1. $\mathfrak{I}, i \models \bigcirc\Phi_1 \vee \bigcirc\Phi_2$ iff $\mathfrak{I}, i \models \bigcirc\Phi_1$ or $\mathfrak{I}, i \models \bigcirc\Phi_2$ iff $i < n$ and ($\mathfrak{I}, i+1 \models \Phi_1$ or $\mathfrak{I}, i+1 \models \Phi_2$) iff $i < n$ and $\mathfrak{I}, i+1 \models \Phi_1 \vee \Phi_2$ iff $\mathfrak{I}, i \models \bigcirc(\Phi_1 \vee \Phi_2)$.

2. $\mathfrak{I}, i \models \Phi_1 \vee (\Phi_2 \wedge \Phi_3)$ iff $\mathfrak{I}, i \models \Phi_1$ or ($\mathfrak{I}, i \models \Phi_2$ and $\mathfrak{I}, i \models \Phi_3$) iff $\mathfrak{I}, i \models (\Phi_1 \vee \Phi_2) \wedge (\Phi_1 \vee \Phi_3)$.

3. $\mathfrak{I}, i \models \Phi_1 \,\mathcal{U}\, \Phi_2$ iff there is a $k \in [i, n]$ s.t. $\mathfrak{I}, k \models \Phi_2$ and $\mathfrak{I}, j \models \Phi_1$ for all $j \in [i, k)$ iff $\mathfrak{I}, i \models \Phi_2$ or $\big(\mathfrak{I}, i \models \Phi_1$, $i < n$, and there is a $k \in [i+1, n]$ s.t. $\mathfrak{I}, k \models \Phi_2$ and $\mathfrak{I}, i \models \Phi_1$ and $\mathfrak{I}, j \models \Phi_1$ for all $j \in [i+1, k)\big)$ iff $\mathfrak{I}, i \models \Phi_2 \vee (\Phi_1 \wedge \bigcirc(\Phi_1 \,\mathcal{U}\, \Phi_2))$.

4. $\mathfrak{I}, i \models \Phi_1 \,\mathcal{R}\, \Phi_2$ iff $\mathfrak{I}, i \models \Phi_2 \wedge (\Phi_1 \vee \blacklozenge(\Phi_1 \,\mathcal{R}\, \Phi_2))$ follows from the above case for $\mathcal{U}$ and the definitions of $\mathcal{R}$ and $\blacklozenge$, i.e., $\Phi_1 \,\mathcal{R}\, \Phi_2 \equiv \neg(\neg\Phi_1 \,\mathcal{U}\, \neg\Phi_2) \equiv \neg(\neg\Phi_2 \vee (\neg\Phi_1 \wedge \bigcirc(\neg\Phi_1 \,\mathcal{U}\, \neg\Phi_2))) \equiv \Phi_2 \wedge (\Phi_1 \vee \blacklozenge(\Phi_1 \,\mathcal{R}\, \Phi_2))$.

5. $\mathfrak{I}, i \models \blacklozenge\Phi$ iff $i = n$ or ($i < n$ and $\mathfrak{I}, i+1 \models \Phi$) iff $\mathfrak{I}, i \models \mathsf{last}$ or $\mathfrak{I}, i \models \bigcirc\Phi$ iff $\mathfrak{I}, i \models \mathsf{last} \vee \bigcirc\Phi$.

Hence, any arbitrary number of applications of the above transformation rules to a TCQ Φ and any of its subformulae always results in a TCQ Φ' s.t. $\mathfrak{I}, 0 \models \Phi$ iff $\mathfrak{I}, 0 \models \Phi'$. □

It remains to show that $\mathsf{Transform}(\Phi)$ achieves the desired normal form, and that, in fact, the inductive combination of answers to subformulae is possible. In the subsequent proof of Lemma 4.2.2, we show both statements.

Proof 4.2.2 (of Lemma 4.2.2 [WJN25])

Let Φ be a TCQ and $\Phi' = \mathsf{Transform}(\Phi)$. The proof has two parts.

Part 1 – Normal Form We first show that $\Phi' = \bigwedge_{k\in\{1,\dots,u\}} \Phi_k \wedge \bigwedge_{l\in\{u+1,\dots,v\}} \Phi_l \wedge \bigwedge_{m\in\{v+1,\dots,w\}} \Phi_m$ where each Φ_k is a UnCQ, each $\Phi_l = \bigcirc\Phi'_l$, and each $\Phi_m = \bigcirc\Phi^1_m \vee \Phi^2_m$, with Φ^2_m being a UnCQ.

First, note that Rule 2 transforms Φ' into a conjunction of disjunctions (since Φ is already in NNF). Moreover, Rules 3 and 4 wrap any temporal operator in some $\bigcirc$-formula, i.e., all subformulae of Φ' are either non-temporal or are contained in some $\bigcirc$. We thus

have a conjunction of a) non-temporal disjunctions, i.e., UnCQs, b) temporal formulae (that are wrapped in some $\bigcirc$-formula), or c) disjunctions that contain a temporal formula somewhere. Cases a) and b) directly comply with the above normal form.

It remains to show for the disjunctions of case c) to always be the form $\bigcirc\Psi_1' \vee \Psi_2$ with Ψ_2 being a UnCQ (given the commutativity of $\vee$, we disregard the case $\Psi_2 \vee \bigcirc\Psi_1'$).

For this, let $\Psi_1 \vee \Psi_2$ be a subformula of Φ' not occurring in some $\bigcirc$-formulae and containing a temporal operator. We assume w.l.o.g. the temporal operator to be in Ψ_1 and below show that $\Psi_1 = \bigcirc\Psi_1'$ for some TCQ Ψ_1'. By this, we directly infer that Ψ_2 cannot be temporal: If Ψ_2 were temporal, by above argumentation, $\Psi_2 = \bigcirc\Psi_2'$, and Rule 1 would have been applicable, contradicting that $\Psi_1 \vee \Psi_2$ is a subformula of Φ'. By Rule 2, this non-temporal formula must be a UnCQ. We must hence now show that $\Psi_1 = \bigcirc\Psi_1'$ for a TCQ Ψ_1'.

Since Ψ_1 is in NNF and contains a temporal operator, it can neither be a negation nor a CQ. First, consider the following possibilities on the shape of Ψ_1: $\Psi_1' \wedge \Psi_1''$, $\Psi_1' \,\mathcal{U} \Psi_1''$, or $\Psi_1' \,\mathcal{R} \Psi_1''$. In all cases, Φ' has applicable rewriting Rules (2, 3, or 4, respectively), contradicting the premise.

It remains to show that if $\Psi_1 = \Psi_1' \vee \Psi_1''$, then $\Psi_1' = \bigcirc\hat{\Psi}_1'$. In this case, associativity of $\vee$ can be used to rewrite $\Psi_1 \vee \Psi_2$ to $\bigcirc\hat{\Psi}_1' \vee (\Psi_1'' \vee \Psi_2)$, showing the statement.

We do a proof by structural induction. Let Ψ_1 be of the form $\Psi_1' \vee \Psi_1''$ and contain a temporal operator (which we assume w.l.o.g to be in Ψ_1'). For the base case, Ψ_1' can have the shape $\bigcirc\varphi_1$, $\varphi_1 \,\mathcal{U} \varphi_2$, or $\varphi_1 \,\mathcal{R} \varphi_2$ for CQs φ_1, φ_2. For the latter two cases, Rule 3 and 4 are still applicable, respectively, which contradicts that Ψ_1 is a subformula of Φ'. Thus, $\Psi_1' = \bigcirc\varphi_1$. For the induction step, Ψ_1' cannot be of the form $\Theta_1 \wedge \Theta_2$, $\Theta_1 \,\mathcal{U} \Theta_2$, or $\Theta_1 \,\mathcal{R} \Theta_2$, for TCQs Θ_1, Θ_2, as this contradicts that $\Psi_1 \vee \Psi_2$ is a subformula of Φ' (Rules 2, 3, or 4 are applicable, respectively). Hence, $\Psi_1' = \bigcirc\Theta_1$, in which case the statement is shown, or $\Psi_1' = \Theta_1 \vee \Theta_2$, in which case we apply the induction hypothesis and associativity of $\vee$ to transform Ψ_1 to $\bigcirc\Theta_1' \vee (\Theta_2 \vee \Psi_1'')$ for a TCQ Θ_1'.

Part 2 – Inductive Combination Based on the now proven normal form, it remains to show that

$$\begin{aligned}\mathsf{cert}_{\mathcal{K}_{\geq j}}(\Phi') = \textstyle\bigcap_{k\in\{1,\dots,u\}} \mathsf{cert}_{K_j}(\Phi_k) \cap \\ \textstyle\bigcap_{l\in\{u+1,\dots,v\}} \mathsf{cert}_{K_{\geq j+1}}(\Phi_l') \cap \\ \textstyle\bigcap_{m\in\{v+1,\dots,w\}} \mathsf{cert}_{K_{\geq j+1}}(\Phi_m^1) \cup \mathsf{cert}_{K_j}(\Phi_m^2).\end{aligned}$$

For the first two conjunctions, this amounts to applying semantics of $\wedge$ and $\bigcirc$ to the previously established shape of Φ'. Note that the semantics of $\bigcirc$ requires $j < n$ for satisfaction, which coincides with our convention that $\mathsf{cert}_{K_{\geq j+1}}(\cdot) = \emptyset$ if $j \geq n$. For the last conjunction, $\mathsf{cert}_{\mathcal{K}_{\geq j+1}}(\Phi_m^1) \cup \mathsf{cert}_{\mathcal{K}_j}(\Phi_m^2)$, we can safely combine answers to both disjuncts, as they refer to disjoint times j and $j+1, \dots, n$. Therefor, we require respecting

the constant domain assumption of Definition 3.4.2, i.e., ensuring answers to different time points to be over the same domain. As shown in Proof 4.1.1, this holds up to $\mathcal{SRIQ}^{(\mathcal{D})}$.□

Answering Unions of Possibly Negated CQs

After showing how Step 1, i.e., the rewriting step, of the motivating example can be provably correctly generalized, it remains to compute the base case, i.e., answering non-temporal queries (Steps 2 and 3 of the motivating example). Recall that we must answer UnCQs of the form $\bigvee_{i=1,\dots,k} \varphi_i \vee \bigvee_{j=k+1,\dots,l} \neg\varphi_j$ over a non-temporal KB $\mathcal{K}$ as the base case. The algorithm for this, called $\mathsf{AnswerUnCQ}(\mathcal{K}, \Psi)$, is defined via the established way of reduction to UCQ entailment [BBL13]. As entailment of UnCQs is equivalent to satisfiability of conjunctions of possibly negated CQs, we can reuse the results of Lemma 4.1.2 (b). Specifically, Section 4.1 introduced the function $\mathsf{Sat}^{\mathcal{K}}_{\Phi}(Z)$ that returns true iff there is a model $\mathcal{I}$ of $\mathcal{K}$ s.t. $\mathcal{I} \models \varphi$ for every $\varphi \in Z$, and $\mathcal{I} \not\models \varphi$ for every $\varphi \notin Z$. For answering a UnCQ $\Psi(\vec{x}) = \bigvee_{i=1,\dots,k} \varphi_i \vee \bigvee_{j=k+1,\dots,l} \neg\varphi_j$, we can hence define $\mathsf{AnswerUnCQ}(\mathcal{K}, \Psi(\vec{x})) := \{\vec{a} \in \mathsf{Ind}(\mathcal{K})^{|\vec{x}|} \mid \mathsf{Sat}^{\mathcal{K}}_{\Psi(\vec{a})}(\{\varphi_{k+1}(\vec{a}), \dots, \varphi_l(\vec{a})\}) \text{ is false}\}$, with $\varphi_h \notin Z$ iff $h \in \{1, \dots, k\}$. The following shows this to be correct.

Lemma 4.2.3 ([WJN25])

For a KB $\mathcal{K}$ and a UnCQ $\Psi(\vec{x})$ it holds that $\mathsf{AnswerUnCQ}(\Psi(\vec{x}), \mathcal{K}) = \mathsf{cert}_{\mathcal{K}}(\Psi(\vec{x}))$.

Proof 4.2.3 (of Lemma 4.2.3 [WJN25])

The Lemma follows directly from the definitions above and Lemma 4.1.2.

$$
\begin{aligned}
\mathsf{AnswerUnCQ}(\mathcal{K}, \Psi(\vec{x})) &= \{\vec{a} \in \mathsf{Ind}(\mathcal{K})^{|\vec{x}|} \mid \mathsf{Sat}^{\mathcal{K}}_{\Psi(\vec{a})}(\{\varphi_{k+1}(\vec{a}), \dots, \varphi_l(\vec{a})\}) \text{ is false}\} \\
\textit{(per Lemma 4.1.2)} &= \{\vec{a} \in \mathsf{Ind}(\mathcal{K})^{|\vec{x}|} \mid \text{there is no } \mathcal{I} \text{ s.t. } \mathcal{I} \models \mathcal{K}, \\
&\qquad\qquad \mathcal{I} \not\models \varphi_1(\vec{a}), \dots, \mathcal{I} \not\models \varphi_k(\vec{a}), \\
&\qquad\qquad \mathcal{I} \models \varphi_{k+1}(\vec{a}), \dots, \mathcal{I} \models \varphi_l(\vec{a})\} \\
&= \{\vec{a} \in \mathsf{Ind}(\mathcal{K})^{|\vec{x}|} \mid \text{for all } \mathcal{I} \text{ it holds that if } \mathcal{I} \models \mathcal{K} \text{ then one of} \\
&\qquad\qquad \mathcal{I} \models \varphi_1(\vec{a}), \dots, \mathcal{I} \models \varphi_k(\vec{a}), \\
&\qquad\qquad \mathcal{I} \not\models \varphi_{k+1}(\vec{a}), \dots, \mathcal{I} \not\models \varphi_l(\vec{a}) \text{ holds}\} \\
&= \{\vec{a} \in \mathsf{Ind}(\mathcal{K})^{|\vec{x}|} \mid \mathcal{K} \models \Psi(\vec{a})\} \\
&= \mathsf{cert}_{\mathcal{K}}(\Psi(\vec{x}))
\end{aligned}
$$

□

Main Algorithm

We are now ready to derive the main algorithm for answering TCQs based on the introduced rewriting. For this, Algorithm 3 – originally presented by Westhofen et al.

[WJN25] – implements the recursive formula of Lemma 4.2.2. Recall that the Lemma allows a safe decomposition of the set of certain answers for a given TCQ in NNF.

Algorithm 3 $\mathsf{AnswerTCQ}(\mathcal{K}, \Phi(\vec{x}), i)$ [WJN25]

Input: a TKB $\mathcal{K} = (\mathcal{O}, (\mathcal{D})_{i\in\{0,\dots,n\}})$, a TCQ $\Phi(\vec{x})$ in NNF, $i \in \mathbb{N}$
Output: $\mathsf{cert}_{\mathcal{K}_{\geq i}}(\Phi(\vec{x}))$

1: **if** $i > n$ **then return** $\emptyset$ **end if**
2: $C \coloneqq \mathsf{Ind}(\mathcal{K})^{|\mathsf{AVar}(\Phi(\vec{x}))|}$
3: $\Phi' \coloneqq \mathsf{Transform}(\Phi(\vec{x}))$ // Φ' in CNF
4: **for all** conjuncts Ψ in Φ' **do**
5: **if** $\Psi = \bigcirc\Psi'$ **then**
6: $D \coloneqq \mathsf{AnswerTCQ}(\mathcal{K}, \Psi, i+1)$
7: **else if** $\Psi = \Psi_1 \vee \bigcirc\Psi_2$ or $\Psi = \bigcirc\Psi_2 \vee \Psi_1$ **then**
8: $D \coloneqq \mathsf{AnswerUnCQ}(\mathcal{K}_i, \Psi_1) \cup \mathsf{AnswerTCQ}(\mathcal{K}, \Psi_2, i+1)$ // Ψ_1 non-temporal
9: **else**
10: $D \coloneqq \mathsf{AnswerUnCQ}(\mathcal{K}_i, \Psi)$ // Ψ non-temporal
11: **end if**
12: $C \coloneqq C \cap D$
13: **end for**
14: **return** C

We start by rewriting the query in Line 3, which results in a TCQ Φ' in CNF. This allows iterating over all conjuncts Ψ in Line 4 and combining their answers later in Line 12. Per Lemma 4.2.2, each Ψ is either a $\bigcirc$-formula – in which case we advance to the next time point in Line 6 – a disjunction with one disjunct being non-temporal and the other a $\bigcirc$-formula – in which case we can combine the answers to both disjuncts as per Lemma 4.2.2 in Line 8 – or a UnCQ – in which case we determine $\mathsf{cert}_{\mathcal{K}_i}(\Psi)$ by the procedure given for Lemma 4.2.3 in Line 10. Since Algorithm 3 allows computing the set of certain answers for a given time point $i \in \mathbb{N}$, we must invoke the algorithm for $i = 0$ to compute $\mathsf{cert}_{\mathcal{K}}(\Phi)$.

Example 4.2.1

For illustration of our procedure, we use the running example $\Phi_{ex}(x) = \Box\mathtt{Human}(x) \wedge \Diamond\mathtt{Pedestrian}(x) \equiv \mathsf{false}\,\mathcal{R}\mathtt{Human}(x) \wedge \mathsf{true}\,\mathcal{U}\mathtt{Pedestrian}(x)$, for which the set of certain answers is $\mathsf{cert}_{\mathcal{K}_t}(\Phi) = \{\mathtt{Jon}, \mathtt{Jane}\}$. Algorithm 3 starts by applying Rules 3 and 4 on $\mathsf{true}\,\mathcal{U}\mathtt{Pedestrian}(x)$ and $\mathsf{false}\,\mathcal{R}\mathtt{Human}(x)$, continues to resolve the resulting disjunctions by Rule 2, and finally uses Rule 5 to eliminate $\bullet$. The result is $\Phi' = \mathtt{Human}(x) \wedge (\bigcirc(\mathsf{false}\,\mathcal{R}\mathtt{Human}(x)) \vee \mathsf{last}) \wedge (\bigcirc(\mathsf{true}\,\mathcal{U}\mathtt{Pedestrian}(x)) \vee \mathtt{Pedestrian}(x))$.

We can now safely combine answers to the non-temporal UnCQs from AnswerUnCQ, as both temporal disjunctions $\bigcirc(\mathsf{false}\,\mathcal{R}\mathtt{Human}(x)) \vee \mathsf{last}$ and $\bigcirc(\mathsf{true}\,\mathcal{U}\mathtt{Pedestrian}(x)) \vee$

$\texttt{Pedestrian}(x)$ adhere to the normal form $\bigcirc\Phi_1 \vee \Phi_2$ with Φ_2 being non-temporal. For this, we compute $\mathsf{cert}_{(\mathcal{K}_t)_0}(\texttt{Human}(x)) = \{\texttt{Jane}, \texttt{Jon}\}$, $\mathsf{cert}_{(\mathcal{K}_t)_0}(\texttt{Pedestrian}(x)) = \emptyset$, and $\mathsf{cert}_{(\mathcal{K}_t)_0}(\mathsf{last}) = \emptyset$. Advancing to $(\mathcal{K}_t)_1$ recursively descends into the $\bigcirc$-subformulae of Φ', i.e., $\mathsf{false}\,\mathcal{R}\texttt{Human}(x)$ and $\mathsf{true}\,\mathcal{U}\texttt{Pedestrian}(x)$, for which the above steps are repeated, with the difference that $\mathsf{cert}_{(\mathcal{K}_t)_1}(\texttt{Pedestrian}(x)) = \{\texttt{Jane}, \texttt{Jon}\}$. The inductive combination in Line 12 leads to the result $\mathsf{cert}_{\mathcal{K}_t}(\Phi) = \{\texttt{Jane}, \texttt{Jon}\}$.

For proving the correctness of Algorithm 3 as stated in Theorem 4.2.1, Lemma 4.2.2 and Lemma 4.2.3 serve as the central building blocks.

Theorem 4.2.1 (Correctness of Algorithm 3 [WJN25])

For a TCQ $\Phi(\vec{x})$ and TKB $\mathcal{K} = (\mathcal{O}, (\mathcal{D}_i)_{i\in\{0,\dots,n\}})$ over a DL between $\mathcal{ALC}$ and $\mathcal{SRIQ}^{(\mathcal{D})}$, it holds that $\mathsf{AnswerTCQ}(\mathcal{K}, \Phi(\vec{x}), 0) = \mathsf{cert}_{\mathcal{K}}(\Phi(\vec{x}))$.

Proof 4.2.4 (of Theorem 4.2.1 [WJN25])

Let $\mathcal{K} = (\mathcal{O}, (\mathcal{D})_{i\in\{0,\dots,n\}})$ be a TKB and Φ be a TCQ. We show the more general statement that, for some $j \leq n$, $\mathsf{AnswerTCQ}(\mathcal{K}, \Phi, j) = \mathsf{cert}_{\mathcal{K}_{\geq j}}(\Phi)$. Recall that the base case of Algorithm 3 is defined for $k > n$ as $\mathsf{AnswerTCQ}(\mathcal{K}, \Phi, k) = \emptyset$.

If $j \leq n$, $\mathsf{Transform}(\Phi) =: \Phi'$ ensures that $\Phi \equiv \Phi'$ and that Φ' is in the normal form given in Lemma 4.2.2. Since Lemma 4.2.2 also yields that

$$\mathsf{cert}_{\mathcal{K}_{\geq j}}(\Phi') = \bigcap_{k\in\{1,\dots,u\}} \mathsf{cert}_{\mathcal{K}_j}(\Phi_k) \cap \bigcap_{l\in\{u+1,\dots,v\}} \mathsf{cert}_{\mathcal{K}_{\geq j+1}}(\Phi'_l) \cap \bigcap_{m\in\{v+1,\dots,w\}} \left(\mathsf{cert}_{\mathcal{K}_{\geq j+1}}(\Phi^1_m) \cup \mathsf{cert}_{\mathcal{K}_j}(\Phi^2_m)\right),$$

it remains to show that Algorithm 3 applies this recursive definition. However, this is straightforward: First, it loops through all conjuncts and intersects their answers. Second, for each conjunct, the following equivalences hold. For any

1. UnCQ Φ_k, $\mathsf{AnswerUnCQ}(\mathcal{K}_j, \Phi_k) = \mathsf{cert}_{\mathcal{K}_j}(\Phi_k)$,
2. $\bigcirc\Phi'_l$, $\mathsf{AnswerTCQ}(\mathcal{K}, \Phi'_l, j+1) = \mathsf{cert}_{\mathcal{K}_{\geq j+1}}(\Phi'_l)$ if $j < n$ or else $\emptyset$, and
3. $\bigcirc\Phi^1_m \vee \Phi^2_m$,
   a) $\mathsf{AnswerTCQ}(\Phi^1_m) = \mathsf{cert}_{\mathcal{K}_{\geq j+1}}(\Phi^1_m)$ if $j < n$ or else $\emptyset$, and
   b) $\mathsf{AnswerUnCQ}(\Phi^2_m) = \mathsf{cert}_{\mathcal{K}_j}(\Phi^2_m)$.

Cases 1. and 3. (b) are directly implied by Lemma 4.2.3. Cases 2. and 3. (a) follow inductively from the recursive definition of Algorithm 3, including the above-mentioned base case, and the correctness of the previous cases. Therefore, $\mathsf{AnswerTCQ}(\mathcal{K}, \Phi', j) = \mathsf{cert}_{\mathcal{K}_{\geq j}}(\Phi')$, and, as $\Phi \equiv \Phi'$, $\mathsf{AnswerTCQ}(\mathcal{K}, \Phi, j) = \mathsf{cert}_{\mathcal{K}_{\geq j}}(\Phi)$. □

4.2.2 Computational Complexity

While this new approach has now shown to be correct, it remains to determine its computation complexity. As Theorem 3.4.1 has shown, TCQ entailment over $\mathcal{ALC}$ is ExpTime-complete. As Theorem 4.2.2 states, the rewriting-based algorithm runs in exponential and hence optimal time.

Theorem 4.2.2 ([WJN25])

For a TCQ Φ and an $\mathcal{ALC}$-TKB $\mathcal{K}$, Algorithm 3 runs in exponential time.

Proof 4.2.5 (of Theorem 4.2.2 [WJN25])

From the Proof of Lemma 4.2.2, we know that $\mathsf{Transform}(\Phi)$ produces a TCQ of the form $\Phi' = \bigwedge_{k\in\{1,\dots,u\}} \Phi_k \wedge \bigwedge_{l\in\{u+1,\dots,v\}} \Phi_l \wedge \bigwedge_{m\in\{v+1,\dots,w\}} \Phi_m$ where each Φ_k is a UnCQ, each $\Phi_l = \bigcirc\Phi'_l$, and each $\Phi_m = \bigcirc\Phi^1_m \vee \Phi^2_m$, with Φ^2_m being a UnCQ. Φ' contains at most exponentially many conjuncts. Specifically, $w \leq 2^{3n}$, where n is the number of subformulae of Φ: Rule 1 does not increase the number of subformulae. Rules 3 and 4 can only be applied once on each subformula (as afterwards, the temporal formula is wrapped in a next operator and will not be considered further) and create, at worst, three new subformulae. Rule 5 introduces an additional disjunction, but is also limited to one application as last is atomic and the other disjunct is temporal. Since none of these rules can be applied twice to the same formula, after applying rules 1, 3, 4, and 5 until none of them is applicable anymore, we have only linearly increased the number of subformulae in Φ. Additionally, it is known that rule 2 creates a CNF (recall that we assume NNF), which can be at most exponential in size. It follows that $w \leq 2^{3n}$.

For each $i \in \{1,\dots,w\}$, the algorithm now handles Φ_i by calling $\mathsf{AnswerUnCQ}$ on Φ_i ($i \in \{1,\dots,u\}$), advancing to the next time point ($i \in \{u+1,\dots,v\}$), or both ($i \in \{v+1,\dots,w\}$). As each Φ_i is linear in the size of Φ, at each time, we do, in the worst case, exponentially many checks of linearly sized UnCQs.

It remains to show that, for a Boolean UnCQ $\Psi = \bigvee_{i=1,\dots,k} \varphi_i \vee \bigvee_{j=k+1,\dots,l} \neg\varphi_j$ and an $\mathcal{ALC}$-KB $\mathcal{K}$, $\mathsf{AnswerUnCQ}(\Psi,\mathcal{K})$ runs in exponential time as well. In the Boolean case, $\mathsf{AnswerUnCQ}(\Psi,\mathcal{K})$ amounts to checking $\mathsf{Sat}^{\mathcal{K}}_{\Psi}(\{\phi_{k+1},\dots,\phi_l\})$. For this procedure, an exponential run time has already been shown in Proof 4.1.4. We can hence conclude that $\mathsf{AnswerTCQ}$ runs in exponential time as well. □

The optimality also holds for more expressive DLs such as $\mathcal{ALCI}$, where query entailment is 2ExpTime-complete [Lut07] and hence $\mathsf{AnswerTCQ}(\mathcal{K},\Phi,0)$ runs in double-exponential time. Essentially, as Algorithm 3 performs an exponential number of entailment checks, it inherits the complexity of the underlying query entailment problem, if this problem is ExpTime-hard. We now consider a case where this does not hold, i.e., where query entailment has a lower complexity.

While the rewriting-based algorithm is optimal w.r.t. computational complexity, Algorithm 3 has another notable advantage over Algorithm 1: If Algorithm 3 is adapted accordingly, it can offer optimal run time in Horn DLs. Horn DLs are the intersection of DLs with the Horn fragment of first-order logic (where only Horn clauses are allowed). They form a popular class of DLs as standard reasoning tasks often become more tractable [KRH13]. In this class, expressiveness is reduced by disallowing arbitrary disjunctions and negations. An example is DL-Lite$_{\text{horn}}$: Here, only axioms of the form $\bigsqcap_{i\in\{0,\dots,n\}} B_i \sqsubseteq B$ are allowed in the ontology, where B_i and B can be formed according to $\bot \mid \mathtt{C} \mid \geq q\mathtt{r}$, for $\mathtt{C} \in \mathsf{N_C}$, $q \in \mathbb{N}$, and $\mathtt{r} \in \mathsf{N_R}$.

These restrictions allow inductively combining the answers to disjuncts, i.e., $\mathsf{cert}_{\mathcal{K}}(\varphi_1 \vee \varphi_2) = \mathsf{cert}_{\mathcal{K}}(\varphi_1) \cup \mathsf{cert}_{\mathcal{K}}(\varphi_2)$. To adapt Algorithm 3 for this setting, we can disregard Rules 1, 2, and 5 and use only the expansion laws of Rules 3 and 4. Since the conjunction of disjunctions is no longer given (since the removed rules establish a CNF), we can simply implement the recursive semantics of Definition 3.4.5 instead of the main loop over all conjuncts. By this, we achieve an algorithm that uses only polynomially many calls to the query engine, since, at each time point, the produced formula is linear in the size of the TCQ Φ. For each of the linearly many CQs, we require one answering operation. For DL-Lite$_{\text{horn}}$, this is an NP-complete problem [Art+09]. Hence, with the above approach for TCQ answering over DL-Lite$_{\text{horn}}$ we get an algorithm with a polynomial run time as well, i.e., temporal operators can be added "for free." In contrast, the automaton-based approach is, independent of the expressiveness of the underlying DL, exponential due to the FA construction [GV13].

4.3 Comparison of Both Approaches

It remains to further analyze the foundational difference between both algorithms. An empirical evaluation follows in Chapter 5; this section provides a theoretical analysis for an improved understanding of the practical differences obtained in the subsequent evaluation. We earlier argued that software implementations will very likely rely on *deterministic* FA due to eager minimization during intermediate construction steps. In order to be able to later on explain the observed differences of the implemented algorithms, this examination must therefore assume Algorithm 1 to rely on deterministic FA.

The essential difference between the algorithms based on rewriting and deterministic FA lies in the emergence of UCQs. To understand this closer, consider the example query $\Phi_{ex}(x) = \Box\mathtt{Human}(\mathtt{Jon}) \wedge \Diamond\mathtt{Pedestrian}(\mathtt{Jon})$ over the TKB $\mathcal{K}_t$ given in Example 3.4.1, for which the introduction to Section 4.2 already showed that Algorithm 1 uses unnecessary UCQs if used in combination with deterministic FA. UCQ checks have no implications on theoretical complexity, but CQ answering is highly optimizable, e.g., through "atom-by-atom" examination relying on syntactical analyses of the completion graph obtained

by the initial consistency check [Sir06]. CQ answering must therefore be preferred, if possible. Recall that we were able to perform the following translation, which shows that no UCQs must be checked on the example:

$$\mathcal{K}_t \models \Box\texttt{Human(Jon)} \wedge \Diamond\texttt{Pedestrian(Jon)} \iff$$
$$\mathcal{K}_t \models (\texttt{Human(Jon)} \wedge \bigcirc\texttt{Human(Jon)}) \wedge (\texttt{Pedestrian(Jon)} \vee \bigcirc\texttt{Pedestrian(Jon)})$$

However, the resulting deterministic FA for $\Box\texttt{Human(Jon)} \wedge \Diamond\texttt{Pedestrian(Jon)}$ (depicted in Figure 4.2) introduces two UCQs: It asks for satisfiability of $\texttt{Human(Jon)} \wedge \neg\texttt{Pedestrian(Jon)}$ and $\texttt{Human(Jon)} \wedge \texttt{Pedestrian(Jon)}$. This translates to checking non-entailment of $\neg\texttt{Human(Jon)} \vee \texttt{Pedestrian(Jon)}$ and $\neg\texttt{Human(Jon)} \vee \neg\texttt{Pedestrian(Jon)}$.

To understand the origin of these UCQs deeper, let us from here on consider only the case of KBs of length two, meaning that we can expand the example query to $\Phi = (\texttt{Human(Jon)} \wedge \bigcirc\texttt{Human(Jon)}) \wedge (\texttt{Pedestrian(Jon)} \vee \bigcirc\texttt{Pedestrian(Jon)})$. Then, the automaton $A_{\mathsf{PA}(\neg\Phi)}$, given in Figure 4.4 encodes checking satisfiability of

$$\neg\Big(\big((\underline{\texttt{Human(Jon)} \wedge \texttt{Pedestrian(Jon)}}) \wedge \bigcirc\texttt{Human(Jon)}\big) \vee$$
$$\big(\underline{\texttt{Human(Jon)} \wedge \neg\texttt{Pedestrian(Jon)}}\big) \wedge \bigcirc(\texttt{Human(Jon)} \wedge \texttt{Pedestrian(Jon)})\big)\Big).$$

The problematic satisfiability checks of conjunctions of CQs are from here on underlined.

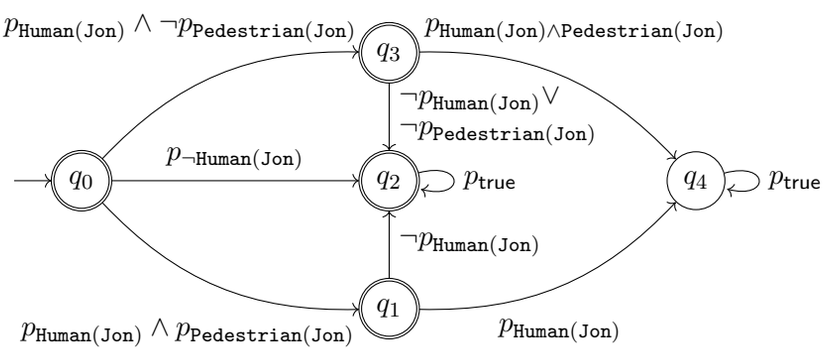

Figure 4.4: Deterministic FA created for the satisfiability check of $\neg((\texttt{Human(Jon)} \wedge \bigcirc\texttt{Human(Jon)}) \wedge (\texttt{Pedestrian(Jon)} \vee \bigcirc\texttt{Pedestrian(Jon)}))$, representing its malignant but equivalent reformulation $\neg(((\underline{\texttt{Human(Jon)} \wedge \texttt{Pedestrian(Jon)}}) \wedge \bigcirc\texttt{Human(Jon)}) \vee (\underline{\texttt{Human(Jon)} \wedge \neg\texttt{Pedestrian(Jon)}}) \wedge \bigcirc(\texttt{Human(Jon)} \wedge \texttt{Pedestrian(Jon)}))$.

By examining the deterministic FA, the origin of the UCQs becomes evident: During deterministic FA construction, additional conjuncts must be introduced to uniquely distinguish between the first and second disjunct of Φ. In that way, it is not possible

for both disjuncts to be satisfied at the same time, and, in the propositional case, the successor for q_0 is always unique. Note that this property does not hold for DLs, but, for reasons explained in Section 4.1.1, the approach of Section 4.1 is based on FA that are deterministic in the propositional case.

Let us now compare these insights to the rewriting approach. In fact, the formula represented by the automaton (which has UCQs) is equivalent – except for the outmost negation – to Φ, which is the formula checked by the rewriting approach (which does not have UCQs):

$$
\begin{aligned}
&\neg\Big(\big((\underline{\texttt{Human}(\texttt{Jon}) \wedge \texttt{Pedestrian}(\texttt{Jon})}) \wedge \bigcirc\texttt{Human}(\texttt{Jon})\big)\vee \\
&\quad\big((\underline{\texttt{Human}(\texttt{Jon}) \wedge \neg\texttt{Pedestrian}(\texttt{Jon})}) \wedge \bigcirc(\texttt{Human}(\texttt{Jon}) \wedge \texttt{Pedestrian}(\texttt{Jon}))\big)\Big) \\
\equiv\ &\big(\neg\texttt{Human}(\texttt{Jon}) \vee \neg\texttt{Pedestrian}(\texttt{Jon}) \vee \neg\bigcirc\texttt{a}(\texttt{Jon})\big)\wedge \\
&\big(\neg\texttt{Human}(\texttt{Jon}) \vee \texttt{Pedestrian}(\texttt{Jon}) \vee \neg\bigcirc\texttt{Human}(\texttt{Jon}) \vee \neg\bigcirc\texttt{Pedestrian}(\texttt{Jon})\big) \\
\equiv\ &\big(\neg\texttt{Human}(\texttt{Jon}) \vee \neg\bigcirc\texttt{Human}(\texttt{Jon})\big)\vee \\
&\big(\neg\texttt{Pedestrian}(\texttt{Jon}) \wedge (\texttt{Pedestrian}(\texttt{Jon}) \vee \neg\bigcirc\texttt{Pedestrian}(\texttt{Jon}))\big) \\
\equiv\ &\big(\neg\texttt{Human}(\texttt{Jon}) \vee \neg\bigcirc\texttt{Human}(\texttt{Jon})\big) \vee \big(\neg\texttt{Pedestrian}(\texttt{Jon}) \wedge \neg\bigcirc\texttt{Pedestrian}(\texttt{Jon})\big) \\
\equiv\ &\neg\Big(\big(\texttt{Human}(\texttt{Jon}) \wedge \bigcirc\texttt{Human}(\texttt{Jon})\big) \wedge \big(\texttt{Pedestrian}(\texttt{Jon}) \vee \bigcirc\texttt{Pedestrian}(\texttt{Jon})\big)\Big) \\
=\ &\neg\Phi
\end{aligned}
$$

To show the equivalence, we push the outmost negation inwards by de Morgan's laws, and are then able to factor out the common disjunction $\neg\texttt{Human}(\texttt{Jon}) \vee \neg\bigcirc\texttt{Human}(\texttt{Jon})$. Finally, the negation can be pushed outward again, leading to $\neg\Phi$.

It is therefore possible to find benign reformulations of $\neg\Phi$ for which satisfiability checking does not require UCQ answering. For example, an automaton could be constructed to represent the satisfiability check of $(\neg\texttt{Human}(\texttt{Jon})\vee\neg\bigcirc\texttt{Human}(\texttt{Jon}))\vee(\neg\texttt{Pedestrian}(\texttt{Jon})\wedge \neg\bigcirc\texttt{Pedestrian}(\texttt{Jon}))$, which has no UCQs. In fact, it is possible to represent this benign query by a nondeterministic FA, as depicted in Figure 4.5.

We must therefore conclude that, in the analyzed example, the additional UCQs are introduced because of the dependency on deterministic FA. This unfortunately leads to entailment checks of UCQs in the DL setting.

The canonical question is whether the example of Figure 4.5 can be generalized: Is it always due to determinization that unnecessary UCQs checks are performed? Or, more formally: For queries where the rewriting approach does not require UCQ answering on a given TKB, is there a corresponding nondeterministic FA without conjunctions on its edges that can be used on the TKB instead?

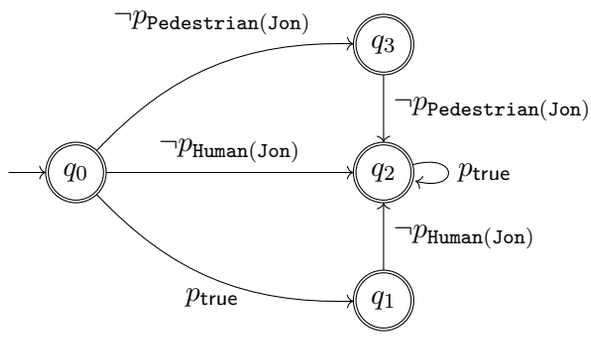

Figure 4.5: Nondeterministic FA created for the satisfiability check of $\neg((\texttt{Human}(\texttt{Jon}) \wedge \bigcirc\texttt{Human}(\texttt{Jon})) \wedge (\texttt{Pedestrian}(\texttt{Jon}) \vee \bigcirc\texttt{Pedestrian}(\texttt{Jon})))$, representing its benign and equivalent reformulation $(\neg\texttt{Human}(\texttt{Jon}) \vee \neg\bigcirc\texttt{Human}(\texttt{Jon})) \vee (\neg\texttt{Pedestrian}(\texttt{Jon}) \wedge \neg\bigcirc\texttt{Pedestrian}(\texttt{Jon}))$.

4.3.1 TCQ_{CQ}: A UCQ-Free Fragment

For answering whether UCQs arise due to determinization, we start by characterizing the class of queries for which UCQ answering is not required. Note that, in general, avoiding UCQ answering is not possible for expressive DLs. Yet, there are some queries which can be answered by using only CQs, e.g., the example query $\Phi_{ex}(x) = \Box\texttt{Human}(x) \wedge \Diamond\texttt{Pedestrian}(x)$, as demonstrated previously. In fact, this is possible for a range of TCQ classes, including "safety" ($\Box\varphi_1$), "liveness" ($\Diamond\varphi_1$), "one-off until" ($(\bigcirc\varphi_1)\,\mathcal{U}\varphi_2$), and conjunctions thereof, for CQs φ_1, φ_2.

However, it is not always directly evident whether a query is checkable by only CQs. For example, the seemingly benign query $\Phi_b = \Diamond(\texttt{A}(\texttt{a}) \wedge \bigcirc\texttt{B}(\texttt{a}))$ must involve UCQ answering. Consider $\mathcal{K}_b = (\{\top \sqsubseteq \texttt{A} \sqcup \texttt{B}\}, (\{\texttt{A}(\texttt{a})\}, \emptyset, \{\texttt{B}(\texttt{a})\}))$ for which, counter-intuitively, $\mathcal{K}_b \models \Phi_b$. Essentially, $\texttt{A}(\texttt{a}) \vee \texttt{B}(\texttt{a})$ has to be checked at the second time point originating from combining $\Diamond$ and $\bigcirc$. While it is easy to see that a reduction to CQ answering is possible for non-trivial queries, Φ_b highlights that this does not have to be immediately obvious.

We start by giving a generic formal characterization of the subset of all TCQs that can be answered by CQs.

Definition 4.3.1 (TCQ_{CQ} [WJN25])

A TCQ Φ is in TCQ_{CQ} if there is an algorithm that answers Φ over arbitrary TKBs $\mathcal{K}$ and whose only access to the data is provided by a CQ answering oracle for the non-temporal KBs $\mathcal{K}_j$.

Note that Definition 4.3.1 is a semantic one, and it is not obvious how TCQs in this fragment might be shaped. It is thus unclear how (and whether) an algorithm can be

devised for deciding membership to $\mathrm{TCQ}_{\mathrm{CQ}}$. $\mathrm{TCQ}_{\mathrm{CQ}}$ is surely not obviously decidable, given that Rice's theorem implies that checking whether an algorithm acts as the "witness" mentioned in Definition 4.3.1 is undecidable.

A detailed examination of the decidability of $\mathrm{TCQ}_{\mathrm{CQ}}$ is gladly not required. Our goal is to identify a nontrivial fragment of TCQs with two properties:

1. Algorithm 3 does not use UCQ checks on the fragment, and
2. queries in this fragment are syntactically characterized, allowing us to generate a nontrivial set of examples on which the deterministic FA induces UCQ checks.

The semantic nature of Definition 4.3.1 impedes these two goals: It is unclear whether Algorithm 3 shows indeed optimal behavior w.r.t. UCQs on *all* TCQs (which is what has to be shown for the first property). For the second property, no syntactic characterization appears to be immediately derivable from Definition 4.3.1. We therefore instead identify a syntactic fragment $\mathrm{TCQ}_{\mathrm{CQ}^*}$ of $\mathrm{TCQ}_{\mathrm{CQ}}$ for which membership can be decided purely based on the syntax of the query. This allows us to easily generate a list of many practically relevant example queries for which UCQ checks are not required, but the corresponding deterministic FA implies them anyway.

For specifying the desired syntactic fragment, for a TCQ Φ, we define $\mathsf{T}(\Phi)$ – the time points the query refers to – as

- $\mathsf{T}(\mathsf{true}) = \mathsf{T}(\mathsf{false}) = \mathsf{T}(\mathsf{last}) = \emptyset$,
- $\mathsf{T}(\varphi) = \{0\}$ for any CQ φ,
- $\mathsf{T}(\Phi_1 \vee \Phi_2) = \mathsf{T}(\Phi_1 \wedge \Phi_2) = \mathsf{T}(\Phi_1) \cup \mathsf{T}(\Phi_2)$,
- $\mathsf{T}(\bullet\Phi') = \mathsf{T}(\bigcirc\Phi') = 1 + \mathsf{T}(\Phi')$, and
- $\mathsf{T}(\Phi_1 \,\mathcal{R}\, \Phi_2) = \mathsf{T}(\Phi_1 \,\mathcal{U}\, \Phi_2) = \mathbb{N} + (\mathsf{T}(\Phi_1) \cup \mathsf{T}(\Phi_2))$.

Here, $1 + A \coloneqq \{1 + a \mid a \in A\}$ and $A + B \coloneqq \{a + b \mid a \in A, b \in B\}$ for sets $A, B \subseteq \mathbb{N}$.

Definition 4.3.2 ($\mathrm{TCQ}_{\mathrm{CQ}^*}$ [WJN25])

Φ is in $\mathrm{TCQ}_{\mathrm{CQ}^*}$ if it does not contain negations and for all subformulae of the form

1. $\Phi_1 \,\mathcal{U}\, \Phi_2$ we have $\mathsf{T}(\Phi_2) \cap (\mathsf{T}(\Phi_1) \cup (1 + \mathsf{T}(\Phi_1 \,\mathcal{U}\, \Phi_2))) = \emptyset$,
2. $\Phi_1 \,\mathcal{R}\, \Phi_2$ we have $\mathsf{T}(\Phi_1) \cap (1 + \mathsf{T}(\Phi_1 \,\mathcal{R}\, \Phi_2)) = \emptyset$,
3. $\Phi_1 \vee \Phi_2$ we have $\mathsf{T}(\Phi_1) \cap \mathsf{T}(\Phi_2) = \emptyset$.

Example 4.3.1

The following queries are members of $\mathrm{TCQ}_{\mathrm{CQ}^*}$, for CQs φ_1, φ_2, and φ_3:

1. $\varphi_1 \vee \bigcirc\varphi_2$ (temporally disjoint disjunctions),
2. $\Box\varphi_1$ (safety),
3. $\Diamond\varphi_1$ (liveness),
4. $\varphi_1 \mathcal{R} \varphi_2$ (release),
5. $\varphi_1 \mathcal{R} (\varphi_2 \mathcal{R} \varphi_3)$ (right-hand nested release),
6. $(\bigcirc\varphi_1) \mathcal{U} \varphi_2$ (one-off until),
7. $\varphi_1 \mathcal{R} ((\bigcirc\varphi_2) \mathcal{U} \varphi_3)$ (release of one-off until),
8. $\Box\Diamond\varphi_1$ (last data entails φ_1),
9. $\Box(\varphi_1 \wedge \bullet\Box\varphi_2)$ (offset of two global constraints), and
10. conjunctions of all of the above.

Membership to $\mathrm{TCQ}_{\mathrm{CQ}^*}$ can be verified by computing the time points the subformulae refer to. For example, for one-off until, $\mathsf{T}(\bigcirc\varphi_1) = \{1\}$ and $\mathsf{T}(\varphi_2) = \{0\}$, which satisfies Constraint 1 of Definition 4.3.2: $\mathsf{T}(\varphi_2) \cap (\mathsf{T}(\bigcirc\varphi_1) \cup (1 + \mathbb{N} + (\mathsf{T}(\bigcirc\varphi_1) \cup \mathsf{T}(\varphi_2)))) = \{0\} \cap (\{1\} \cup \{n \in \mathbb{N} \mid n \geq 1\}) = \emptyset$.

Of the nine specified queries, only the first example and those with only one CQ induce no UCQ checks in their corresponding deterministic FA, as verified with Lydia, a software tool for translating LTL_f formulae to deterministic FA [GF21]. Additionally, on *all* 36 possible conjunctions of above queries (assuming different CQs), UCQs arise. In total, these sum up to 41 out of 45 nontrivial properties in $\mathrm{TCQ}_{\mathrm{CQ}^*}$ that require UCQ checks by Algorithm 1. This corresponds to over 90% of the example queries. Of the remaining four, only one query contains more than one CQ. These examples highlight that the automaton-based algorithm does not behave optimally on $\mathrm{TCQ}_{\mathrm{CQ}^*}$.

Before showing that $\mathrm{TCQ}_{\mathrm{CQ}^*}$ is in fact a fragment of $\mathrm{TCQ}_{\mathrm{CQ}}$, we need a supplementary statement.

Lemma 4.3.1 ([WJN25])

Let $\Phi \in \mathrm{TCQ}_{\mathrm{CQ}^*}$ and $\Psi = \mathsf{Transform}(\Phi)$. Any subformula of Ψ is again in $\mathrm{TCQ}_{\mathrm{CQ}^*}$.

Proof 4.3.1 (of Lemma 4.3.1 [WJN25])

For showing that $\mathrm{TCQ}_{\mathrm{CQ}^*}$ is closed under $\mathsf{Transform}$, let $\Phi \in \mathrm{TCQ}_{\mathrm{CQ}^*}$. We have the following cases when applying $\mathsf{Transform}$ on Φ (note that any subformula of Φ is also in $\mathrm{TCQ}_{\mathrm{CQ}^*}$):

1. $\Phi = \bigcirc\Phi_1 \vee \bigcirc\Phi_2$ thus $\Psi = \bigcirc(\Phi_1 \vee \Phi_2)$: As $\Phi_1, \Phi_2 \in \mathrm{TCQ}_{\mathrm{CQ}^*}$ and $(1 + \mathsf{T}(\Phi_1)) \cap (1 + \mathsf{T}(\Phi_2)) = \emptyset$, it follows that $\mathsf{T}(\Phi_1) \cap \mathsf{T}(\Phi_2) = \emptyset$, and thus $\Psi \in \mathrm{TCQ}_{\mathrm{CQ}^*}$.

2. $\Phi = (\Phi_2 \wedge \Phi_3) \vee \Phi_1$ or $\Phi = \Phi_1 \vee (\Phi_2 \wedge \Phi_3)$ thus $\Psi = (\Phi_1 \vee \Phi_2) \wedge (\Phi_1 \vee \Phi_3)$: As $\Phi_1, \Phi_2, \Phi_3 \in \mathrm{TCQ}_{\mathrm{CQ}^*}$ and $(\mathsf{T}(\Phi_2) \cup \mathsf{T}(\Phi_3)) \cap \mathsf{T}(\Phi_1) = \emptyset$, it follows that $\mathsf{T}(\Phi_1) \cap \mathsf{T}(\Phi_2) = \emptyset$ and $\mathsf{T}(\Phi_1) \cap \mathsf{T}(\Phi_3) = \emptyset$ and thus $\Psi \in \mathrm{TCQ}_{\mathrm{CQ}^*}$.

3. $\Phi = \Phi_1 \mathcal{U} \Phi_2$ thus $\Psi = \Phi_2 \vee (\Phi_1 \wedge \bigcirc(\Phi_1 \mathcal{U} \Phi_2))$: As $\Phi_1, \Phi_2 \in \mathrm{TCQ}_{\mathrm{CQ}^*}$ and $\mathsf{T}(\Phi_2) \cap (\mathsf{T}(\Phi_1) \cup (1 + \mathsf{T}(\Phi_1 \mathcal{U} \Phi_2))) = \emptyset$, it follows that $\mathsf{T}(\Phi_2) \cap \mathsf{T}(\Phi_1) = \emptyset$ and $\mathsf{T}(\Phi_2) \cap \mathsf{T}(\bigcirc(\Phi_1 \mathcal{U} \Phi_2)) = \mathsf{T}(\Phi_2) \cap (1 + \mathsf{T}(\Phi_1 \mathcal{U} \Phi_2)) = \emptyset$ and thus $\Psi \in \mathrm{TCQ}_{\mathrm{CQ}^*}$.

4. $\Phi = \Phi_1 \mathcal{R} \Phi_2$: Analogous to the above case.

5. $\Phi = \blacksquare\Phi'$ thus $\Psi = \mathsf{last} \vee \bigcirc\Phi'$: As $\Phi' \in \mathrm{TCQ}_{\mathrm{CQ}^*}$ and $\mathsf{T}(\mathsf{last}) \cap \mathsf{T}(\bigcirc\Phi') = \emptyset$, it follows that $\Psi \in \mathrm{TCQ}_{\mathrm{CQ}^*}$. □

We can now prove Theorem 4.3.1 – stating the correctness of the syntactic $\mathrm{TCQ}_{\mathrm{CQ}}$ fragment – by showing that for $\Phi \in \mathrm{TCQ}_{\mathrm{CQ}^*}$, $\mathsf{AnswerTCQ}(\Phi, \mathcal{K}, i)$ uses only CQ answering for KB reasoning, i.e., when applying $\mathsf{AnswerUnCQ}$.

Theorem 4.3.1 ([WJN25])

$\mathrm{TCQ}_{\mathrm{CQ}^*} \subsetneq \mathrm{TCQ}_{\mathrm{CQ}}$.

Proof 4.3.2 (of Theorem 4.3.1 [WJN25])

$\mathsf{AnswerTCQ}(\mathcal{K}, \Phi, i)$ encounters only CQs in $\mathsf{AnswerUnCQ}$ if we do not consider recursive calls: As per Lemma 4.3.1, $\mathrm{TCQ}_{\mathrm{CQ}^*}$ is closed under $\mathsf{Transform}$ and thus $\mathsf{Transform}(\Phi) \in \mathrm{TCQ}_{\mathrm{CQ}^*}$. As there is no negation present, and, by Constraint 3 in Definition 4.3.2, $\mathsf{Transform}(\Phi)$ cannot contain a nontemporal disjunction of two CQs, Lines 8 and 10 only encounter CQs. Due to Lemma 4.3.1, the inputs to $\mathsf{AnswerTCQ}$ in Lines 6 and 8 satisfy again the assumption that the query is in $\mathrm{TCQ}_{\mathrm{CQ}^*}$ and thus the above argument is also applicable to the recursive calls at time $i + 1$. Therefore, $\mathrm{TCQ}_{\mathrm{CQ}^*} \subseteq \mathrm{TCQ}_{\mathrm{CQ}}$.

Finally, it is easy to see that $\mathrm{TCQ}_{\mathrm{CQ}^*}$ does not cover the whole class of $\mathrm{TCQ}_{\mathrm{CQ}}$, e.g., $\Diamond(\varphi_1 \mathcal{U} \varphi_2)$ for two CQs φ_1, φ_2 can be answered using a CQ engine (as it is equivalent to $\Diamond\varphi_2$) but is not in $\mathrm{TCQ}_{\mathrm{CQ}^*}$. Hence, $\mathrm{TCQ}_{\mathrm{CQ}^*} \subsetneq \mathrm{TCQ}_{\mathrm{CQ}}$. □

Our main goal with $\mathrm{TCQ}_{\mathrm{CQ}^*}$ is to identify reasons behind the differences in UCQ handling between Algorithm 1 and Algorithm 3, which we will continue in Section 4.3.2. However, the fragment can also be leveraged for guiding users towards efficient query fragments during specification. For such practical scenarios, it is desirable that syntactic membership in $\mathrm{TCQ}_{\mathrm{CQ}^*}$ can be efficiently decided before answering (which is, in general, exponential). Fortunately, membership to the syntactical fragment can be computed in polynomial time and linear space.

Theorem 4.3.2 ([WJN25])

Syntactic membership in $\mathrm{TCQ}_{\mathrm{CQ}^*}$, i.e., $\Phi \in \mathrm{TCQ}_{\mathrm{CQ}^*}$, is decidable in polynomial time and linear space.

Intuitively, calculating t requires one traversal of the formula's syntax tree with operations that take polynomial time on the specific shapes of the induced sets. The constraints of Definition 4.3.2 can be checked in polynomial time as well, which is formally shown in the following proof of Theorem 4.3.2.

Proof 4.3.3 (of Theorem 4.3.2 [WJN25])

We first show that computing $\cup$ and $+$ over t can be done in polynomial time and linear space. Note that t can only produce sets of the form $S \cup N_k$, where $S \subset \mathbb{N}$ is finite and $N_k = \{n \in \mathbb{N} \mid n \geq k\}$ for some $k \in \mathbb{N}$. These sets can be symbolically represented by simply storing its integer k. Then, $(S_1 \cup N_{k_1}) \cup (S_2 \cup N_{k_2}) = (S_1 \cup S_2) \cup N_{\min(k_1,k_2)}$ and $(S_1 \cup N_{k_1}) + (S_2 \cup N_{k_2}) = (S_1 + S_2) \cup N_{\min(k_1+\min(S_2),k_2+\min(S_1),k_1+k_2)}$. Both cases require only operations that can be computed in polynomial time and linear space, as $A \cup B$ can be computed in time $O(|A| \cdot |B|)$ and space $O(|A| + |B|)$. As constraints 1. to 3. of Definition 4.3.2 require, at worst, computing t for all subformulae, we require one traversal of the syntax tree of Φ using a bounded number of operations with linear space and polynomial time complexity per node. It remains to check $(S_1 \cup N_{k_1}) \cap (S_2 \cup N_{k_2}) = \emptyset$, which we do for

- N_{k_1} and N_{k_2} non-empty by returning false,
- N_{k_1} and N_{k_2} empty by returning $S_1 \cap S_2 = \emptyset$, which can be done in $O(|S_1| \cdot |S_2|)$ time and $O(|S_1| + |S_2|)$ space, and
- N_{k_1} non-empty and N_{k_2} empty (or vice versa) by checking $S_1 \cap S_2 = \emptyset$ and $s_2 < k_1$ for all $s_2 \in S_2$ (or vice versa), which can be done in $O(|S_1| \cdot |S_2|)$ time and $O(|S_1| + |S_2|)$ space).

Combining all steps shows the polynomial time and linear space complexity of deciding membership in $\mathrm{TCQ}_{\mathrm{CQ}^*}$. □

As a corollary, Theorem 4.3.2 implies that deciding $\Phi \in \mathrm{TCQ}_{\mathrm{CQ}^*}$ is in $\mathrm{NSpace}(O(n))$, i.e., it can be decided by a nondeterministic Turing machine using linear space. As $\mathrm{NSpace}(O(n))$ is equivalent to the class of context-sensitive languages, $\mathrm{TCQ}_{\mathrm{CQ}^*}$ is context-sensitive as well.

It is, however, not possible to give a context-free grammar for $\mathrm{TCQ}_{\mathrm{CQ}^*}$: Consider the regular expression $r = (\bigcirc^*\varphi \vee \bigcirc^*\varphi) \vee \bigcirc^*\varphi$ for some CQ φ, where $L(r) \subset \mathrm{TCQ}$. If we assume that $\mathrm{TCQ}_{\mathrm{CQ}^*}$ is context-free, then $\mathrm{TCQ}_{\mathrm{CQ}^*} \cap L$ must also be context-free (since context-free languages are closed under intersection with regular languages). However, $\mathrm{TCQ}_{\mathrm{CQ}^*} \cap L$ is not context-free, as it is equivalent to $L_{\neq} = \{(\bigcirc^{n_1}\varphi \vee \bigcirc^{n_2}\varphi) \vee \bigcirc^{n_3}\varphi \mid n_1 \neq n_2 \neq n_3 - n_1\}$, which is not context-free.

For the proof, we denote $\bigcirc^{n_1}$ as the first, $\bigcirc^{n_2}$ as the middle, and $\bigcirc^{n_3}$ as the third part of a word in $L_{\neq}$. Let p be the constant given by Odgen's lemma for context-free languages [Ogd68]. Then, $s = (\bigcirc^{p+p!}\varphi \vee \bigcirc^{p}\varphi) \vee \bigcirc^{2p+2p!}\varphi \in L_{\neq}$ and assume markers are put onto the middle part. By Odgen's lemma, there is a decomposition $s = uvwxy$ with vx having a marked letter, vwx having at most p marked letters, and $uv^nwx^ny \in L_{\neq}$ for all $n \in \mathbb{N}$. If v or x contains a b, it directly follows that $uv^0wx^0y \notin L_{\neq}$, and therefore v and x must contain only $\bigcirc$. Combining this with the fact that v or x contains a marked letter, we have only two cases: First, v is fully contained in the middle part of s (and hence neither v nor x cover the first part of s). Let $i = \frac{p!}{q} + 1 \in \mathbb{N}$ where $q = |vx|_{\bigcirc}$ if x is also contained in the middle part of s and $q = |v|_{\bigcirc}$ otherwise. Then, uv^iwx^iy still contains $p! + p$ times $\bigcirc$ in the first part and $p + q \cdot (i-1) = p + q \cdot \frac{p!}{q} = p! + p$ times $\bigcirc$ in the middle part, violating $L_{\neq}$. In the second case, v is fully contained in the first and x is fully contained in the middle part of s (and hence neither v nor x cover the third part of s). Let $i = \frac{p!}{|xv|_{\bigcirc}} + 1 \in \mathbb{N}$. Then, the constraint $n_1 + n_2 \neq n_3$ is violated by uv^iwx^iy since

$$\begin{aligned} n_1 + n_2 &= p! + p + |v|_{\bigcirc} \cdot (i-1) + p + |x|_{\bigcirc} \cdot (i-1) \\ &= p! + 2p + |xv|_{\bigcirc} \cdot \frac{p!}{|xv|_{\bigcirc}} \\ &= 2p! + 2p = n_3. \end{aligned}$$

In all cases, $uv^iwx^iy \notin L_{\neq}$ and $L_{\neq}$ is not context-free. Hence, $\mathrm{TCQ}_{\mathrm{CQ}^*}$ cannot be context-free as well.

4.3.2 Unnecessary Automaton-Induced UCQ Checks in $\mathrm{TCQ}_{\mathrm{CQ}^*}$

We are now able to show that the unnecessary UCQs of Algorithm 1 are introduced by determinization of the automaton – not only on the example query of the introduction to this section, but on the substantial class of queries in $\mathrm{TCQ}_{\mathrm{CQ}^*}$. The proof is done by giving a nondeterministic FA that can replace the deterministic FA when computing $\mathsf{cert}_{\mathcal{K}}(\Phi)$ for a query $\Phi \in \mathrm{TCQ}_{\mathrm{CQ}^*}$ in Algorithm 1. When checking satisfiability of Φ w.r.t. $\mathcal{K}$ using this nondeterministic FA, no UCQs have to be answered. Any UCQ check

induced by the deterministic FA is hence "unnecessary" in the sense that they do not occur when using its nondeterministic replacement.

We construct the nondeterministic FA $A_{\neg\mathsf{PA}(\Phi),n+1}$ for a given TCQ $\Phi \in \mathrm{TCQ}_{\mathrm{CQ}^*}$ and TKB $\mathcal{K}$ of length $n+1$ that can replace $A_{\neg\mathsf{PA}(\Phi)}$ created in Line 1 of Algorithm 1. The core idea is to leverage the nondeterminism to encode the recursive structure of Algorithm 3 in the automaton. Intuitively, we can apply the rewriting rules up to the required depth $n+1$, negate the query, and encode the resulting Disjunctive Normal Form (DNF) using nondeterministic choices in the automaton. Recall that $\mathsf{Transform}(\Phi) = \bigwedge_{k\in\{1,\dots,u\}} \Phi_k \wedge \bigwedge_{l\in\{u+1,\dots,v\}} \Phi_l \wedge \bigwedge_{m\in\{v+1,\dots,w\}} \Phi_m$, where each Φ_k is a UnCQ, each $\Phi_l = \bigcirc\Phi'_l$, and each $\Phi_m = \bigcirc\Phi^1_m \vee \Phi^2_m$, with Φ^2_m being a UnCQ. More specifically, as Proof 4.3.2 has shown, if $\Phi \in \mathrm{TCQ}_{\mathrm{CQ}^*}$, we can safely assume that Φ_k and Φ^2_m are CQs. $\mathsf{Transform}$ only applies the rewriting rules for the current depth, i.e., it does not apply to formulae wrapped into $\bigcirc$-operators. For creating the automaton later on, we need the fully transformed, negated query up to depth $n+1$. For this, we create $\Psi = \mathsf{FTN}(\Phi, n+1)$ with

$$\mathsf{FTN}(\Phi,i) = \begin{cases} \bigvee_{k\in\{1,\dots,u\}} \neg\Phi_k \vee \bigvee_{l\in\{u+1,\dots,v\}} \CIRCLE\mathsf{FTN}(\Phi'_l, i-1) \vee \bigvee_{m\in\{v+1,\dots,w\}} \left(\CIRCLE\mathsf{FTN}(\Phi^1_m, i-1) \wedge \neg\Phi^2_m\right) \text{ if } i > 1, \\ \bigvee_{k\in\{1,\dots,u\}} \neg\Phi_k \vee \bigvee_{m\in\{v+1,\dots,w\}} \neg\Phi^2_m \text{ if } i = 1, \end{cases}$$

where $\Phi_k, \Phi'_l, \Phi^1_m, \Phi^2_m$ are given according to $\mathsf{Transform}(\Phi)$. For a TCQ Ψ arising from applying FTN, we recursively define its nondeterministic FA $A_{\mathsf{PA}(\Psi)} = (Q^\Psi, \Sigma, q^\Psi_0, \delta^\Psi, F^\Psi)$ over $\Sigma = 2^{\{p_\varphi \mid \varphi\in\mathsf{CQ}(\Psi)\}}$ as

$$A_{\mathsf{PA}(\Psi)} = \begin{cases} (\{q_0,q_1,q_2\} \uplus Q^{\Psi_1} \uplus Q^{\Psi_2}, \Sigma, q_0, \{(q_0,\epsilon,q_1),(q_0,\epsilon,q_2)\} \uplus \delta^{\Psi_1}_{q_0|q_1} \uplus \delta^{\Psi_2}_{q_0|q_2}, \emptyset) \\ \quad \text{if } \Psi = \Psi_1 \vee \Psi_2, \\ (\{q_0,q_1\} \uplus Q^{\Psi_1}, \Sigma, q_0, \Delta(q_0,\neg\Psi_2,q_1) \cup \Delta(q_1,\mathsf{true},q_1) \uplus \delta^{\Psi_1}_{q_0|q_1}, \{q_0,q_1\}) \\ \quad \text{if } \Psi = \CIRCLE\Psi_1 \wedge \neg\Psi_2 \text{ for a UnCQ } \Psi_2, \\ (\{q_0,q_1\} \uplus Q^{\Psi'}, \Sigma, q_0, \Delta(q_0,\mathsf{true},q_1) \uplus \delta^{\Psi'}_{q_0|q_1}, \{q_0,q_1\}) \text{ if } \Psi = \CIRCLE\Psi', \\ (\{q_0,q_1\}, \Sigma, q_0, \Delta(q_0,\neg\Psi',q_1) \cup \Delta(q_1,\mathsf{true},q_1), \{q_0,q_1\}) \\ \quad \text{if } \Psi = \neg\Psi' \text{ for a UnCQ } \Psi'. \end{cases}$$

We use $\Delta(q_0,\neg\Psi,q_1)$ for a UnCQ $\Psi = \bigvee_{g=1,\dots,k} \varphi_g \vee \bigvee_{h=k+1,\dots,l} \neg\varphi_h$ as a shorthand notation for the set of transitions $\{(q_0, \{p_\psi \mid \psi \in (Z \cup \{\varphi_{k+1},\dots,\varphi_l\}) \setminus \{\varphi_1,\dots,\varphi_k\}\}, q_1) \mid Z \subseteq \mathsf{CQ}(\Psi)\}$ and $\Delta(q_0,\mathsf{true},q_1)$ for $\{(q_0,\{p_\psi \mid \psi\in Z\},q_1) \mid Z \subseteq \mathsf{CQ}(\Psi)\}$. Moreover, $\delta^{\Psi'}_{q_0|q_j}$ is $\delta^{\Psi'}$ where $q_0 \in Q^{\Psi'}$ is substituted by $q_j \in Q^\Psi$, i.e., q_j replaces the initial state of $A_{\mathsf{PA}(\Psi')}$ when merging it into $A_{\mathsf{PA}(\Psi)}$.

Example 4.3.2

The nondeterministic replacement FA for the example query $\Phi_{ex}(\mathtt{Jon})$ over a TKB of

length two is given in Figure 4.6. For this, we first determine

$$\begin{aligned}
&\mathsf{FTN}(\Phi_{ex}(\texttt{Jon}), 2)\\
\equiv&\neg\texttt{Human}(\texttt{Jon}) \vee \bullet\mathsf{FTN}(\Box\texttt{Human}(\texttt{Jon}), 1)\vee\\
&(\bullet\mathsf{FTN}(\Diamond\texttt{Pedestrian}(\texttt{Jon}), 1) \wedge \neg\texttt{Pedestrian}(\texttt{Jon}))\\
\equiv&\neg\texttt{Human}(\texttt{Jon}) \vee \bullet\neg\texttt{Human}(\texttt{Jon})\vee\\
&(\bullet\neg\texttt{Pedestrian}(\texttt{Jon}) \wedge \neg\texttt{Pedestrian}(\texttt{Jon})).
\end{aligned}$$

Subsequently, we are able to encode the disjunctive structure of this formula in the transitions of Figure 4.6.

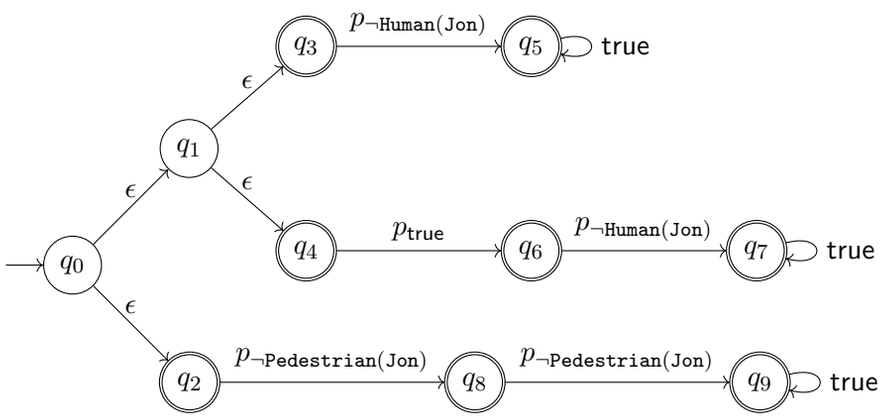

Figure 4.6: The nondeterministic replacement FA $A_{\mathsf{PA}(\mathsf{FTN}(\Phi_{ex}(\texttt{Jon}),2))}$.

If we remove ϵ-transitions and join the sink states q_5, q_7, and q_9, it is homomorphic to the already introduced FA of Figure 4.5.

$A_{\mathsf{PA}(\Psi)}$ does, in fact, not induce UCQ checks, as it contains transitions only of the form ϵ, $\Delta(q_0, \mathsf{true}, q_1)$, or $\Delta(q_0, \neg\varphi, q_1)$. For the first two, no access to the data is required, and for the latter, according to Lemma 4.1.2, the satisfiability of $\Delta(q_0, \neg\varphi, q_1)$ can simply be determined in iteration i by checking $\mathcal{K}_i \not\models \varphi$. It remains to show that $A_{\mathsf{PA}(\Psi)}$ is a correct replacement for $A_{\mathsf{PA}(\neg\Phi)}$ used in Line 1 of Algorithm 1.

Theorem 4.3.3 (Correctness of $A_{\mathsf{PA}(\mathsf{FTN}(\Phi,i))}$)

For a TCQ Φ, $i > 0$, and $w \in \{p_\varphi \mid \varphi \in \mathsf{CQ}(\Phi)\}^i$, it holds that $w \in L(A_{\mathsf{PA}(\neg\Phi)})$ iff $w \in L(A_{\mathsf{PA}(\mathsf{FTN}(\Phi,i))})$.

Note that $i = 0$ is excluded above as Definition 3.4.1 does not allow for databases of zero length and thus any word considered by Algorithm 1 must be non-empty.

Proof 4.3.4 (of Theorem 4.3.3)

Our first task is the correctness of the automaton construction $A_{\mathsf{PA}(\mathsf{FTN}(\Psi,i))}$. The second step is proving that for any temporal interpretation $\mathfrak{I}$ of length $i > 0$ and TCQ Φ, $\mathfrak{I}, 0 \models \neg\Phi$ iff $\mathfrak{I}, 0 \models \mathsf{FTN}(\Phi, i)$. From this, Lemma 4.1.1, and the correctness of both automaton constructions – $A_{\mathsf{PA}(\mathsf{FTN}(\Psi,i))}$ and the standard construction for $\mathrm{LTL_f}$ formulae [GV13] – the statement of Theorem 4.3.3 follows.

We now prove correctness of $A_{\mathsf{PA}(\mathsf{FTN}(\Psi,i))}$, i.e., for a TCQ Φ, $i > 0$, and $w \in \{p_\varphi \mid \varphi \in \mathsf{CQ}(\Phi)\}^i$, $w \in L(A_{\mathsf{PA}(\mathsf{FTN}(\Psi,i))})$ iff $w \models \mathsf{PA}(\mathsf{FTN}(\Psi, i))$, which we do by induction over i. For the base case, $\mathsf{FTN}(\Phi, 1) = \bigvee_{k\in\{1,\dots,u\}} \neg\Phi_k \vee \bigvee_{m\in\{v+1,\dots,w\}} \neg\Phi^2_m$ for UnCQs Φ_k and Φ^2_m, and therefore any $w \models \mathsf{PA}(\mathsf{FTN}(\Psi, 1))$ is either $w = \epsilon$ or $w \subseteq \{p_\varphi \mid \varphi \in \mathsf{CQ}(\Phi)\}$ not satisfying the constraints of some UnCQ Φ_j with $j \in \{1, \dots, u\} \cup \{v + 1, \dots, w\}$. By definition, $A_{\mathsf{PA}(\mathsf{FTN}(\Psi,1))}$ can reach an accepting state for words of length one only by a sequence of ϵ-transitions optionally followed by a transition from $\Delta(\cdot, \neg\Phi_j, \cdot)$ for a UnCQ Φ_j with $j \in \{1, \dots, u\} \cup \{v + 1, \dots, w\}$, which concludes the base case.

For the induction step, consider a word $w = P_1 \dots P_i$ of length $i > 1$. By $\mathrm{LTL_f}$ semantics, $w \models \mathsf{PA}(\mathsf{FTN}(\Phi, i))$ iff

a) $P_1 \not\models \Phi_k$ for some $k \in \{1, \dots, u\}$, or

b) $P_2 \dots P_i \models \mathsf{FTN}(\Phi'_l, i - 1)$ for some $l \in \{u + 1, \dots, v\}$, or

c) $P_2 \dots P_i \models \mathsf{FTN}(\Phi^1_m, i - 1)$ and $P_1 \not\models \Phi^2_m$ for some $m \in \{v + 1, \dots, w\}$

(by considering only words of the same length $i > 1$, $\bullet$ can be treated as $\bigcirc$). On the other hand, by definition, $w \in L(A_{\mathsf{PA}(\mathsf{FTN}(\Psi,i))})$ iff there is an accepting sequence of ϵ-transitions optionally followed by

i) a transition in $\Delta(\cdot, \neg\Phi_k, \cdot)$ optionally followed by transitions in $\Delta(\cdot, \mathsf{true}, \cdot)$, or

ii) a transition in $\Delta(\cdot, \mathsf{true}, \cdot)$, optionally followed by a sequence of transitions to an accepting state in $A_{\mathsf{PA}(\mathsf{FTN}(\Phi'_l,i-1))}$, or

iii) a transition in $\Delta(\cdot, \neg\Phi^2_m, \cdot)$, optionally followed by transitions in $\Delta(\cdot, \mathsf{true}, \cdot)$ followed by a sequence of transitions to an accepting state in $A_{\mathsf{PA}(\mathsf{FTN}(\Phi^1_m,i-1))}$.

for $k \in \{1, \dots, u\}$, $l \in \{u + 1, \dots, v\}$, and $m \in \{v + 1, \dots, w\}$. As cases i), ii), and iii) correspond exactly to the models of the $\mathrm{LTL_f}$ formula described in the cases a), b), and c) above, using the induction hypothesis for $i - 1$ yields the correctness of $A_{\mathsf{PA}(\mathsf{FTN}(\Phi,i))}$.

It remains to show the satisfiability equivalence of $\neg\Phi$ and $\mathsf{FTN}(\Phi, i)$ w.r.t. interpretations of length i. First, note that in Lemma 4.2.1, we have established that $\mathsf{Transform}(\Phi)$ preserves the semantics of Φ. By this and the duality of $\wedge$ and $\vee$ as well as $\bigcirc$ and $\bullet$, we

postulate the following prerequisite equivalences:

$$\mathfrak{I}, 0 \models \neg\Phi \text{ iff } \mathfrak{I}, 0 \models \neg\mathsf{Transform}(\Phi) \text{ iff}$$
$$\mathfrak{I}, 0 \models \bigvee_{k\in\{1,\ldots,u\}} \neg\Phi_k \vee \bigvee_{l\in\{u+1,\ldots,v\}} \bullet\neg\Phi'_l \vee \bigvee_{m\in\{v+1,\ldots,w\}} \bullet\neg\Phi^1_m \wedge \neg\Phi^2_m$$

with Φ_k, Φ^2_m UnCQs.

It remains to show that $\mathfrak{I}, 0 \models \neg\Phi$ iff $\mathfrak{I}, 0 \models \mathsf{FTN}(\Phi, i)$, which we do by induction over i. For the base case, i.e., a temporal interpretation $\mathfrak{I}$ of length $i = 1$, using our prerequisite equivalences, we need to show

$$\mathfrak{I}, 0 \models \bigvee_{k\in\{1,\ldots,u\} m\in\{v+1,\ldots,w\}} \neg\Phi_k \vee \bigvee \neg\Phi^2_m = \mathsf{FTN}(\Phi, 1) \text{ iff}$$
$$\mathfrak{I}, 0 \models \bigvee_{k\in\{1,\ldots,u\}} \neg\Phi_k \vee \bigvee_{l\in\{u+1,\ldots,v\}} \bullet\neg\Phi'_l \vee \bigvee_{m\in\{v+1,\ldots,w\}} \bullet\neg\Phi^1_m \wedge \neg\Phi^2_m.$$

This amounts to showing that $\mathfrak{I}, 0 \models \bullet\neg\Psi'_l$ and $\mathfrak{I}, 0 \models \bullet\neg\Psi^1_m$ is always true, which holds as $\mathfrak{I}$ is a sequence of length one. For the induction step, consider a temporal interpretation $\mathfrak{I}$ of length $i > 1$. Then, by the prerequisite equivalences,

$$\mathfrak{I}, 0 \models \neg\Phi \text{ iff } \mathfrak{I}, 0 \models \bigvee_{k\in\{1,\ldots,u\}} \neg\Phi_k \vee \bigvee_{l\in\{u+1,\ldots,v\}} \bullet\neg\Phi'_l \vee \bigvee_{m\in\{v+1,\ldots,w\}} \bullet\neg\Phi^1_m \wedge \neg\Phi^2_m \text{ and}$$
$$\mathfrak{I}, 0 \models \mathsf{FTN}(\Phi, i) \text{ iff } \mathfrak{I}, 0 \models \bigvee_{k\in\{1,\ldots,u\}} \neg\Phi_k \vee \bigvee_{l\in\{u+1,\ldots,v\}} \bullet\mathsf{FTN}(\Phi'_l, i-1) \vee \bigvee_{m\in\{v+1,\ldots,w\}} \bullet\mathsf{FTN}(\Phi^1_m, i-1) \wedge \neg\Phi^2_m.$$

Thus, showing that $\mathfrak{I}, 0 \models \neg\Phi$ iff $\mathfrak{I}, 0 \models \mathsf{FTN}(\Phi, i)$ amounts to showing that $\mathfrak{I}, 0 \models \mathsf{FTN}(\Phi'_l, i-1)$ iff $\mathfrak{I}, 0 \models \neg\Phi'_l$ and $\mathfrak{I}, 0 \models \mathsf{FTN}(\Phi^1_m, i-1)$ iff $\mathfrak{I}, 0 \models \neg\Phi^1_m$, which is given by the induction hypothesis. □

This proves that the superfluous UCQs on the 41 queries identified in Example 4.3.1 arose from relying on deterministic FAs. As argued earlier, this can hardly be avoided in implementations. Due to the practical implications of UCQ answering, the above considerations yield the hypothesis that Algorithm 3 likely performs better on some queries. Specifically, when using the deterministic FA, unnecessary UCQs not only arise on the introductory example of $\Box\texttt{Human}(\texttt{Jon}) \wedge \Diamond\texttt{Pedestrian}(\texttt{Jon})$, but also on the simple queries like $\varphi_1 \wedge \varphi_2$, for two CQs φ_1 and φ_2. The deterministic FA $A_{\mathsf{PA}(\neg(\varphi_1\wedge\varphi_2))}$ has two symbolically encoded transitions: $\neg p_{\varphi_1} \vee \neg p_{\varphi_2}$ to an accepting state and $p_{\varphi_1} \wedge p_{\varphi_2}$ to a sink. Note that the latter satisfiability check induces UCQ answering. Using a nondeterministic FA, we could encode the transition to the sink as true, thereby avoiding conjunctions of CQs on the transition. It appears that the deterministic FA does not encode what would informally be described as "if $\neg\varphi_1 \vee \neg\varphi_2$ ... else ...," but rather describes "if $\neg\varphi_1 \vee \neg\varphi_2$ ... if $\varphi_1 \wedge \varphi_2$ ...," i.e., the "else" condition explicitly. In practice,

this can be avoided during the traversal of Algorithm 1 by performing the disjunctive satisfiability check first. If the query is satisfiable, we must advance to an accepting state and reject the overall TCQ; if the query is not satisfiable, we must advance to the sink and accept the overall TCQ.

However, this trick is not possible if we are presented with at least three successor states in which two transitions induce UCQ checks. This situation appears on many of the queries given in Example 4.3.1, like $\Box\varphi_1 \wedge \Diamond\varphi_2$, $(\bigcirc\varphi_1)\,\mathcal{U}\varphi_2$, $\Diamond\varphi_1 \wedge \Diamond\varphi_2$, $\varphi_1\,\mathcal{R}\varphi_2$, and $\varphi_1\,\mathcal{R}(\varphi_2\,\mathcal{R}\varphi_3)$, all of which are in $\mathrm{TCQ}_{\mathrm{CQ}^*}$. In this case, just like in state q_0 of Figure 4.4, if the non-UCQ-inducing transition is not satisfiable, it is impossible to determine which of the other successor states should be taken without checking the transitions (to q_1 and q_3 in the example). The subsequent evaluation in Chapter 5 shows that avoiding UCQ checks is desirable also from an empirical perspective.

5 Demonstrating Practical Applicability

Contents

The main statement of the dissertation is that temporal querying in expressive DLs can be made practically applicable, specifically for assessing safety-critical scenarios in road traffic. While we have already theoretical examined the computational complexity of both proposed algorithms, practical contexts often require efficient behavior on realistic inputs; on the other hand, unreasonable edge cases can be discarded.

It is therefore imperative to measure how the algorithms perform in the field. For this, we first require an implementation of both techniques: Topllet, as presented in Section 5.1 and originally published by Westhofen et al. [Wes+24b].

Second, we need realistic, industrially relevant, and controllable benchmarks, which are introduced in Section 5.2. For this, the dissertation introduces the A.U.T.O., which has been developed in a large publicly funded research project together with industry partners: first, as a conceptual model by Scholtes, Westhofen, Turner et al. [Sch+21] and subsequently formalized by Westhofen et al. [Wes+22a] based on artifacts systematically created during a criticality analysis as described by Neurohr, Westhofen, Butz et al. [Neu+21]. While many ontologies have been proposed before, an independent review found that A.U.T.O. is more feature-complete than many competitors proposed in the scientific literature [ZKZ23, Table I]. The employed benchmarks based on A.U.T.O. were first presented in 2024 by Westhofen et al. [Wes+24a].

Results of the experiments with Topllet on the benchmarks, as given in the works presenting both algorithms [Wes+24a; WJN25], are discussed in Section 5.3. Here, a special focus is put on a comparison of the two techniques, as Section 4.3 already gave some hints towards a more greedy behavior of the FA-based approach w.r.t. UCQ answering.

The claim of "practical applicability" not only encompasses measures of performance, but also requires showing that the software is actually useful for practitioners. This topic is examined in Section 5.4, where Topllet is integrated into STARS [Sch+24a], a scenario-based testing framework for measuring test coverage, an essential concept in demonstrating SOTIF. The integration has first been introduced as part of a work on advancing scenario coverage towards an OWA by Westhofen et al. [Wes+26]. It shows that profound implications arise when transitioning from standard temporal logics (which make the CWA) to the OWA of MTCQs. Even under the OWA setting, useful statements on coverage can be computed efficiently. By this, we demonstrate that temporal DL queries provide value also in a broader SOTIF application context.

To summarize, this chapter contributes the following aspects to the scientific state-of-the-art:

- the first publicly available implementation of a temporal querying engine for expressive DL ontologies,
- a large ontology formulated in an expressive DL based on an industrially relevant conceptual model,
- the first publicly available benchmark generator for temporal queries over expressive DL ontologies, and
- evidence for the claim that temporal querying over expressive DLs can, in fact, be made practically applicable, from a performance and usability perspective.

5.1 Topllet: An Implementation

We begin with an overview of the tool developed for answering MTCQs in Section 5.1.1, including the user interface and procedures required by both algorithms under consideration. The implementation details of the automaton- and rewriting-based approaches – such as employed optimizations – are subsequently presented in Section 5.1.2 and Section 5.1.3.

5.1.1 Overview

We implemented TOPLLET on top of OPENLLET version 2.6.6, a non-temporal DL reasoner written in Java. OPENLLET is itself a continuation of the established PELLET reasoner, which was first presented by Parsia and Sirin in 2004 [SP04]. It gained popularity due to its extensive feature set and is used in almost all the related work on DLs presented in Section 2.2.2. Development of PELLET was discontinued in 2017 when focus was shifted to the proprietary knowledge graph platform Stardog. In contrast, OPENLLET continues to be maintained by community efforts.

OPENLLET supports various reasoning tasks over OWL2 ontologies [Sir+07]. This includes consistency checking, tCQ answering, explanation generation, and ontology realization. For TOPLLET, we are specifically interested in the consistency and tCQ features, as they offer the desired foundation for the temporal querying mechanisms. Both algorithms were integrated as a novel module into TOPLLET, accessing functionality of other modules containing the basic reasoning services.

The software tool runs on any host machine capable of executing a Java Virtual Machine (JVM) and is compatible with JVMs of version 17 and above. To assure quality of the software, a test suite for the novel implementations is distributed alongside, consisting of over 140 individual test cases. A copy of the tool can be found online [Wes26e].

Usage

The main task of TOPLLET is to answer an MTCQ over a user-supplied TKB, i.e., ontology and data. It can be conveniently accessed via the command line:

```
topllet [-c CATALOG] <QUERY> <TKB>
```

`QUERY` is a plain text file containing an MTCQ, and `TKB` is a plain text file consisting of a list of file paths to non-temporal OWL2-files. Optionally, `CATALOG` can point to an OASIS XML catalog, allowing for custom mappings between ontology Unique Resource Identifiers (URIs) and file locations. This can aid in referencing local file copies in order to reduce loading times. Upon completion, TOPLLET prints a tabular result overview with query variables as columns and answers as rows.

TOPLLET also offers a streaming mode, which can be invoked similar to the default offline setting:

```
topllet [-c CATALOG] <QUERY> <ONTOLOGY>
```

Through a ZeroMQ port, data can be sent or updated as the application produces it. After concluding a time point by a corresponding message, TOPLLET will compute the set of certain answers and return a completion notice. Data updates sent during computation will be discarded (TOPLLET does not buffer them). The application can finally ask for the query results, if desired. Here, instead of a TKB, a non-temporal ontology (OWL2-file) is given. This ontology can then be filled and updated by data sent via the ZeroMQ port. Currently, MTCQ answering over expressive DLs does not lend itself well for complex real-time settings, and it is therefore advised to use only small ontologies and data for the streaming mode.

As introduced above, TOPLLET has two major inputs: a query and a TKB. For representing MTCQs, the software implements an extension of a previously published standard grammar for $\mathrm{LTL_f}$ [Fav20], which is simply a textual representation of the temporal operators, e.g., $\Box$ becomes `G`. In place of atomic propositions, the input language then allows for CQs of the form `:C(?x) & ... & :r(?x,y)`, where answer variables (like `x`) are prefixed by a question mark. Existentially quantified variables (such as `y`) do not have a trailing question mark. We allow users to employ SPARQL-like prefixes to differentiate between ontology namespaces given as URIs. In the notation above, `:` denotes the default, empty namespace.

Any MTCQ input has to be tree-shaped, as we rely on the integrated tCQ answering engine of OPENLLET. In a nutshell, if we view each CQ in the MTCQ as a graph with the relations being edges and variables being nodes, the graph must be tree-shaped (cf. Definition 3.2.25). This subclass of MTCQs is – from a practical point of view – sufficient for most of the relevant properties, as violations of the tree-shape have to occur in the existentially quantified variables. Practical observations show that such variables play only a supplementary role and are used sparingly. Additionally, for many DLs, including $\mathcal{SRIQ}$, a tree-shape violation can simply be resolved by replacing one of the conflicting variables by an individual, as described in Section 3.2.2.

TKBs are represented using a custom format, as no prior proposals exist for combining expressive OWL2 ontologies with temporal data. To keep compatibility with OWL2, TOPLLET implements a straightforward mechanism: Any TKB is represented as a list of paths to OWL2-files, where list positions correspond to discrete time points. Each OWL2-file contains the database of that time point. All files must contain the same imports, i.e., the non-temporal ontologies, which are again OWL2-files.

Besides the command line interface, TOPLLET also provides an Application Programming Interface (API), which can be used for integration into other tools. We use this API in Section 5.4 to add MTCQ answering to a third-party scenario-based testing framework. An example of the API for solving the simple MTCQ $\Box\mathtt{A}(x)$ on the TKB $(\{\mathtt{B} \sqsubseteq \mathtt{A}\}, (\{\mathtt{B(a)}\}, \{\mathtt{B(a)}\}))$ is given in the following:

```
// TKB
TemporalKnowledgeBase tkb = new InMemoryTemporalKnowledgeBaseImpl();
tkb.add(new KnowledgeBaseImpl());
tkb.add(new KnowledgeBaseImpl());
// Ontology
for (KnowledgeBase kb : tkb)
{
  kb.addClass(term("A"));
  kb.addClass(term("B"));
  kb.addSubClass(term("B"), term("A"));
}
// Data
for (KnowledgeBase kb : tkb)
{
  kb.addIndividual(term("a"));
  kb.addType(term("a"), term("B"));
}

// MTCQ
String formula = "G(A(?x))";
MetricTemporalConjunctiveQuery mtcq =
    MetricTemporalConjunctiveQueryParser.parse(formula, tkb);

// Answering
System.out.println("Answering MTCQ " + formula);
MTCQNormalFormEngine eng = new MTCQNormalFormEngine(); // Use the rewriting engine
QueryResult res = eng.exec(mtcq);
System.out.println(res); // Prints "[{FunVar(x)=a}]"
```

Architecture and Shared Components

Topllet extends Openllet by an additional module – `openllet-mtcq` – which implements the core functionality required for Algorithm 1 and Algorithm 3, such as the MTCQ parser and data model, TKB and FA data classes, and the required query engines. Various existing (partially modified) modules are required by `openllet-mtcq`, especially:

- `openllet-cli`: Adds the above described command line interfaces of Topllet.
- `openllet-owlapi`: For loading KBs, while enabling incorporation of OASIS XML catalogs.
- `openllet-query`: Offers highly efficient CQ answering as well as a custom UCQs engine. The query result data class has been modified for increased efficiency. The query model was completely rewritten with generics to enable different query types with low code redundancy.
- `openllet-core`: Access to the consistency check interface for UCQ entailment, which has been modified to allow concept membership checks for multiple individuals at once.

Together with external dependencies, these modules are part of the general architecture of Topllet shown in Figure 5.1.

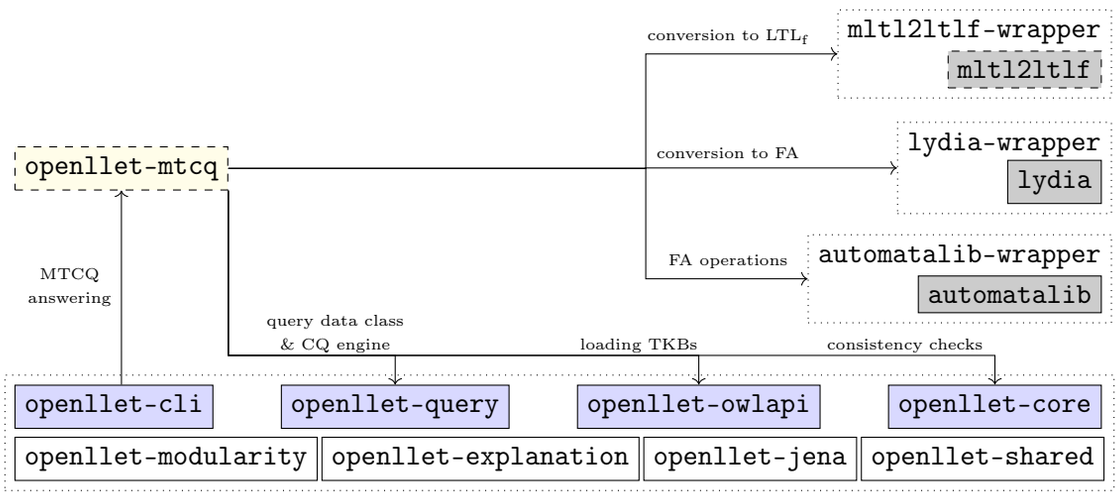

Figure 5.1: Essential modules of Topllet, with external dependencies (gray), new developments (dashed), and modified modules of Openllet (blue). Not all modules are shown.

For the automaton-based approach, Topllet adds three dependencies also depicted in Figure 5.1. First, MLTL2LTLf, a custom Python tool developed for translating MLTL to LTL$_f$. It implements the translation described in Section 3.4.3 and is available online [Wes25a]. Additionally, Lydia offers a translation from LTL$_f$ into FAs [GF21]. We finally require AutomataLib for representing and iterating over the resulting FA [IHS15], which has been extended by a custom class offering convenience methods.

Two changes to Openllet mandate further discussion: how UCQ answering has been implemented and the way query result operations were made more efficient.

UCQ Answering Both MTCQ algorithms require a procedure for answering UCQs, which is not available in Openllet. We hence implemented the algorithm described by Horrocks and Tessaris [HT00] (for UCQs containing tCQs). This engine transforms any UCQ into a CNF, which is then answered one-by-one through transformation to data assertions. To handle undistinguished variables for tree-shaped queries, it relies on the rolling-up procedure, which is already implemented in Openllet. Several optimizations lead to practically applicability: We can inform the UCQ engine about results from the MTCQ engine s.t. it must only answer over a subset of the whole search space. Additionally, the CQ engine is used whenever possible. For example, when CQs of the UCQ operate over disjoint variables, the UCQ can be answered be combining the results

from the CQ engine. A similar consideration can be made for entailment: if the CQ engine finds some disjunct entailed, no expensive UCQ entailment check is required. Otherwise, the CNF is built, where we can similarly terminate early if we find one conjunct not to be entailed.

Query Results While temporal query answering often requires many operations on intermediate query results, optimizations can have drastic effects. In OpenLLet, a query result is a set of so-called bindings, where a binding is a map from variables to individuals. Several operations are possible: (batch) adding and removing elements, inversion, intersection, membership and equality check, and size computation.

We improved efficiency mainly by adding two features: symbolic inversion and partial bindings. Inversion is often performed when leveraging the duality between entailment and satisfiability, and expensive considering that in practice only few individuals are entailed. All mentioned operations regard symbolic inversion information. Partial bindings are required when examining CQs that contain only parts of all variables of the MTCQ. Here, explicitly enumerating all satisfiable bindings for non-constrained variables of a CQ too early would be needlessly wasteful. Hence, we allow storing bindings that only map a part of the variables to individuals and avoid early expansion. This helps, for example, when checking for membership of bindings referring to the same subset of all variables – as done, e.g., in determining whether the first run of the automaton-based implementation excluded some (partial) binding. The Cartesian product of the remaining variables is only computed when necessary, e.g., on iteration, adding, or removal.

Without the above explained methods, a large share of run time was spent on query result operations on our benchmarks. While several other improvements could be made (such as using decision diagrams for fully symbolic representation), we found that the two improvements made query result operations negligible overall.

5.1.2 Automaton-Based Algorithm

The automaton-based algorithm implements the following, abstract workflow:

a) parse the MTCQ and build the propositional abstraction of its negation,

b) convert the output of a) to an $\mathrm{LTL_f}$ formula using MLTL2LTLf,

c) convert the $\mathrm{LTL_f}$ formula to an FA using Lydia,

d) parse the FA using AutomataLib,

e) parse the TKB,

f) perform an under-approximating first traversal of the FA using solely the CQ engine, and

g) if necessary, perform a full run where checking edge satisfiability involves b UCQ answering.

The FA is traversed by reading the current data and computing the satisfiable answers to each edge of each currently active state. A candidate that has an accepting run is a counterexample to entailment, i.e., the algorithm returns all answers to which no accepting run was found.

In Section 4.1, we have seen that the algorithm runs in doubly exponential time for deterministic FAs. To achieve adequate run time in practice, the implementation uses the CQ engine in a first traversal as follows. The class `EdgeConstraintChecker` can answer FA edges for satisfiability, which has a toggle for performing either an under-approximating or a full check. Internally, it uses a `SatisfiabilityKnowledgeManager`, which globally manages satisfiability knowledge for queries obtained from checking FA edges. Recall that we are confronted with BCQs, i.e., conjunctions of possibly negated CQs. In order to avoid redundant query answering, the manager computes and caches both satisfiable (and unsatisfiable, if under-approximating) answers for queries at each time point and BCQ. Knowledge is propagated according to Table 4.1: After generating certain answers to a CQ, the manager checks whether the answers can be used as satisfiability results for *any* of the managed BCQs (and not only the currently checked FA edge).

5.1.3 Rewriting-Based Algorithm

The workflow of the rewriting-based implementation can be described similarly. Instead of the recursive formulation made in Section 4.2, Topllet implements an iterative version (for practical performance reasons) as follows:

a) parse the MTCQ,

b) parse the TKB,

c) iterate over each time point and

   i) transform the MTCQ into normal form,

   ii) reorder the resulting CNF,

   iii) answer each non-temporal MTCQ,

   iv) mark temporal MTCQs for the next iteration, and

d) combine all answers.

Note that the iterative procedure differs slightly from the recursive Algorithm 3: It requires keeping a list of MTCQs that have to be examined in the next iteration. Additionally, query results must be stored in a tree-like data structure during iteration for final assembly.

Besides relying on CQ answering wherever possible, we again implement several techniques to further improve performance: First, the CNF is reordered s.t. non-temporal queries which can be answered by the CQ engine appear first, followed the remaining non-temporal and finally temporal queries. Note that more powerful query reordering techniques can be applied [Sir06, Section 6.4], which TOPLLET does not yet make use of on the level of MTCQs. Reordering allows generating candidate sets while examining conjuncts, and we need to examine only the possible answers returned for the previous conjuncts. For the first query in the CNF, the candidate set is inherited from its parent formula. A query cache similar to what has been implemented for the automaton-based algorithm stores intermediate results; however, it now also must respect candidate sets. These can be passed when querying the cache, which then delivers cached results even if it just contains information on a part of the candidates. Only the remaining candidates, as returned by the query cache, need to be checked. As a further optimization, observe that we might descend into $\bigcirc$-subformulae containing the same MTCQ at different places. The iterative approach allows merging those engine calls into one (possibly by extending an already existing candidate set). Finally, we re-use already combined results as much as possible and attach them to the parse tree of the MTCQ. A lazy evaluation of the gathered query results is performed only at the very end.

An important yet separate optimization aspect concerns simplification of the formulae resulting from rewriting. In contrast to the automaton-based procedure, we do not operate on a guaranteed minimal temporal structure. Indeed, practical observations showed that without simplification formulae can grow excessively large, impeding performance specifically due to establishing a CNF. Take, for example, the formula

$$\Phi = (\Diamond\varphi_1) \vee \big(\Diamond(\varphi_2 \wedge \Diamond\varphi_1)\big).$$

By strictly applying the rewriting rules of Figure 4.3, the resulting transformed formula is

$$\Phi' = (\varphi_1 \vee \varphi_2 \vee \bigcirc\Phi) \wedge (\varphi_1 \vee \bigcirc\Phi),$$

now containing Φ twice. Iteratively repeating this procedure on Φ' for each time point $i \in \{0, \ldots, n\}$ leads to an exponential blow up: If we unroll Φ to depth $i+1$, this yields $3 \cdot \sum_{j=0}^{i} 2^j = 3 \cdot (2^{i+1} - 1)$ leaf nodes in the resulting rewritten formula (counting only leafs containing φ_1 or φ_2).

As a first measure, query results are cached and re-used whenever possible. Still, this leads to an exponential size of the data structure storing the intermediate query results, which later must be collapsed into a single set. While caching can also be applied here, it is often more efficient to avoid superfluous – especially exponential – formula sizes in the first place. The following simplification rules are implemented in TOPLLET:

- Idempotence

(Id1) $\Phi \wedge \Phi \rightsquigarrow \Phi$

(Id2) $\Phi \vee \Phi \rightsquigarrow \Phi$

- Self inverse

(In) $\neg\neg\Phi \rightsquigarrow \Phi$

- Tautology

(Ta1) $\Phi \vee \mathsf{true} \rightsquigarrow \mathsf{true}$

(Ta2) $\Phi \vee \neg\Phi \rightsquigarrow \mathsf{true}$

- Contradiction

(Co1) $\Phi \wedge \mathsf{false} \rightsquigarrow \mathsf{false}$

(Co2) $\Phi \wedge \neg\Phi \rightsquigarrow \mathsf{false}$

- Absorption

(Ab1) $\Phi \wedge \mathsf{true} \rightsquigarrow \Phi$

(Ab2) $\Phi \vee \mathsf{false} \rightsquigarrow \Phi$

(Ab3) $(\Phi_1 \wedge \Phi_2) \vee \Phi_1 \rightsquigarrow \Phi_1$

(Ab4) $(\Phi_1 \vee \Phi_2) \wedge \Phi_1 \rightsquigarrow \Phi_1$

- Next aggregation

(Ne1) $\bigcirc\Phi_1 \wedge \bigcirc\Phi_2 \rightsquigarrow \bigcirc(\Phi_1 \wedge \Phi_2)$

(Ne2) $(\bigcirc\Phi_1 \vee \Phi_2) \wedge (\bigcirc\Phi_3 \vee \Phi_2) \rightsquigarrow \bigcirc(\Phi_1 \wedge \Phi_3) \vee \Phi_2$ (if formula is in normal form)

(Ne3) $(\bigcirc\Phi_1 \vee \Phi_2) \wedge (\bigcirc\Phi_1 \vee \Phi_3) \rightsquigarrow \bigcirc\Phi_1 \vee (\Phi_2 \wedge \Phi_3)$ (if formula is in normal form)

These rules are applied until no rule can be fired anymore (under consideration of commutativity and associativity), after any intermediate step of the rewriting procedure. A necessary remark concerns (Ne2) and (Ne3). While a more general formulation can be made, e.g., $(\Phi_1 \vee \Phi_2) \wedge (\Phi_3 \vee \Phi_2) \rightsquigarrow (\Phi_1 \wedge \Phi_3) \vee \Phi_2$ for (Ne2), this would establish a DNF which counteracts the required CNF. However, (Ne2) and (Ne3) collapse two disjuncts already in the required normal form into one disjunct of the same shape, thus retaining the normal form.

Topllet correctly reduces the example Φ' to $\varphi_1 \vee \bigcirc\Phi$ by applying rule (Ab4), avoiding the exponential blow up. Since Φ is equivalent to $\Diamond\varphi_1$, we could have also added $(\Diamond\varphi_1) \vee \big(\Diamond(\varphi_2 \wedge \Diamond\varphi_1)\big) \rightsquigarrow \Diamond\varphi_1$ as a separate simplification rule, however, the more general rules above handle this case on the rewritten query.

5.2 Benchmarks

Ideally, we would be able to use controlled real-world data for the experimental performance evaluation. In Section 5.4, we use drone data from German intersections [Boc+20] in our experiments. For the systematic evaluation of this section, such real world data unfortunately do not offer the required scalability. This calls for synthetic yet realistic benchmark data that can be randomized, scaled in size, and are freely available for replicability. Ideally, we would have two categories of benchmarks: One that can be examined easily and is ideal for investigating technical hyptoheses in a controlled manner and allows for efficient experimentation of how specific configurations affect performance during development. Second, a benchmark that is complex and transfers well to real-world settings and thereby allowing an evaluation of whether Topllet can handle practical performance demands.

We are currently not aware of *any* public benchmark data on querying TKBs. The same was noted by the developers of MeTeoR [Wan+22], where data of the Lehigh University Benchmark [GPH05] are extended with random intervals to enable an evaluation on the OWL2 RL fragment of the benchmark. Unfortunately, a random extension of a non-temporal benchmark might not reflect actual temporal data, e.g., in continuity of concepts over time, and thus might not transfer to real-world applications. This section hence presents two benchmarks:

- the two-wheeler benchmark, which is a lightweight yet scalable temporal querying benchmark for efficient evaluation, and
- the TOBM, a large-scale benchmark generator for scenarios of automated driving applications that closely mimics real-world data.

Additionally, the first benchmark will guide us throughout an easily understandable set of example inputs to Topllet as well as its results.

5.2.1 The Two-Wheeler Benchmark

The two-wheeler benchmark is concerned with a single query Φ_0 for granting right of way to two wheelers and provides TKBs with just three time points with scaling parameter N denoting the number of two-wheelers [Wes+24b]. The benchmark is available online under an MIT license [Wes26f].

We will use this very simple benchmark to ease into using Topllet by means of an example. A typical use case for Topllet is to specify challenging situations for ADSs and check for their presence in data. Of specific interest are traffic rules. Therefore, the two-wheeler benchmark is concerned with the correctly granting right of way to bicyclists or mopeds on crossings. Three inputs to Topllet are required: a) an OWL2 ontology, b) temporal data as a list of references to the data points, and c) a temporal query file.

Ontology

For formalizing the traffic rule, we first gather relevant domain knowledge. As we aim to keep the benchmark lean, we assume only the following three facts:

1. Crossings are equivalent to everything that connects at least three lanes.

2. Every two-wheeled vehicle is a bicycle or moped.

3. Everything that is left of something is also to the side thereof.

In the OWL2 functional-style syntax, our example ontology is formalized as:

1. `EquivalentClasses(Crossing ObjectMinCardinality(3 connects Road))`

2. `SubClassOf(TwoWheeledVehicle ObjectUnionOf(Bicycle Moped))`

3. `SubPropertyOf(toTheLeftOf toTheSideOf)`

This ontology uses qualified cardinality restrictions ($\mathcal{Q}$) and role hierarchies ($\mathcal{R}$), and is therefore contained in the DL $\mathcal{ALCRQ}$. It moreover uses a disjunction on the right-hand side of a concept inclusion (the second axiom) and is hence prone to the issues arising from disjunctions discussed in Chapter 4. Note that trikes are a good example as to why this inclusion may not hold in the other direction – there are mopeds that might not be two-wheeled vehicles.

Data

For the data, operation of the ADS yields a finite time trace of non-temporal data. Each element in the sequence represents a discrete time point by assigning classes to and binary relations between individuals.

The perception of the ADS identified five individuals: the system `s`, a two-wheeled vehicle `t`, and a space `c` connecting the distinct roads `r0`, `r1`, and `r2`, with `r1` being orthogonal and to the right of `r0`. Note that the ADS only perceived a `TwoWheeledVehicle` – possibly, the object classification was not trained on more specific classes. Over three time points t_0, t_1, and t_2, the following relations hold:

t_0: `s` is to the left of and parallel to `t`, `s` and `t` intersect `r0`;

t_1: `s` intersects `r0`, `t` intersects `c`;

t_2: `s` intersects `r1`.

In short, `s` lets `t` move onto the crossing before turning right onto `r1`.

In order to gain scalability for our experiments later on, we add a parameter N, where the above case is $N = 1$, and each additional increment of N probabilistically adds

another two-wheeler by taking a seed S as input. With a configurable probability (by default, 90%), the two-wheeler show the same behavior as above. Otherwise, it always stays on `r0`. For example, $N = 30$ with seed $S = 0$ has, on average, 115 assertions per time point.

Query

The final component is the query, representing a traffic rule: "*Anyone who intends to turn on a crossing shall let bicycles or mopeds enter the intersection first if they drive parallel to them*," which is adapted from section 9 paragraph 1 sentence 2 of the German traffic act. Intuitively, the behavior described in the data above is in accordance with this rule; however, if a two-wheeler stays on `r0` (which can happen with a probability of 10%), the traffic rule is violated. In TOPLLET, we use an MTCQ with four answer variables to represent this behavior as an implication, which we call Φ_0

```
G # fixes the global types of the answers
(
  (:System(?x) & :Road(?q) & :Road(?p) & :orthogonal(?q, ?p))
    &
  ((:Moped(?y)) | (:Bicycle(?y)))
)
  &
(
  F
  (
    (:intersects(?x,?p) & :toTheSideOf(?x,?y) & :parallel(?x,?y))
      &
    F_[0,20] (:intersects(?x,?q))
  ) # if, somewhen, x turns from p to q with y being parallel to it before
  ->
  ( # then, x has to be on p until y is on the crossing (grant right of way)
    (:intersects(?x,?p))
      U
    (:intersects(?y,c) & :Crossing(c) & :connects(c,?p) & :connects(c,?q))
  )
)
```

The query asks for individuals `x` and `y` in the data s.t. `x` is a system that, if it is in a crossing situation with a two-wheeler `y` being parallel and to the side of `x`, grant right of way to `y` before turning. In the given data from above, the set of certain answers returned by TOPLLET contains only one mapping assigning `s` to x, `r0` to p, `r1` to q, and `t` to y.

5.2.2 The Traffic Ontology Benchmark

The second benchmark family, TOBM [Wes+24a], models pedestrians, bicyclists, and vehicles over 200 time points on an X- and a T-crossing with four MTCQs using the A.U.T.O., a large $\mathcal{SRIQ}^{(\mathcal{D})}$ ontology. In contrast to the two-wheeler benchmark, the goal

here is to match a real-world application context as closely as possible. The benchmark is available online under an MIT license [Wes26d].

The Automotive Urban Traffic Ontology

The A.U.T.O. [Wes+22a, Section 5] has been created as part of a large publicly funded research project in a scientific-industrial collaboration. It is a conglomerate of $\mathcal{SRIQ}^{(\mathcal{D})}$ ontologies for the traffic domain and related fields, and currently consists of 1 449 axioms over 676 concepts and 213 roles. A.U.T.O. was already successfully used for analyzing real-world traffic data from drone recordings [Wes+22a, Section 8]. It has been found to outperform a large share of the competition in terms of modeling options [ZKZ23, Table I]. It has also been acknowledged in recent literature [MP25; Mar+25]. Two independent lines of work also extend the ontology: Researchers at Oxford used an adapted version to derive scene graphs on the ROAD dataset [Pan+25]. A group at the Tianjin University extended A.U.T.O. with additional domain knowledge [Zha+24].

Conceptually, A.U.T.O. is based on the Six-Layer Model (6LM) [Sch+21]. The 6LM categorizes traffic entities and their properties into six individual layers:

- Layer 1: road network and traffic guidance objects, which encompasses road structure (roads, lanes, intersections, etc.) and objects designated for guiding traffic (signs, markings, traffic lights, etc.).
- Layer 2: roadside structures, where objects near the road network yet not relevant for guiding traffic are contained (such as buildings and vegetation).
- Layer 3: temporary modifications of the former, describing intentional or unintentional alterations of static objects, e.g., construction elements or alterations due to destructive forces such as storms.
- Layer 4: dynamic objects, including all traffic participant and their vehicles, but also animals and inanimate moving objects.
- Layer 5: environmental conditions, such as weather, time, and illumination information.
- Layer 6: digital information, specifically, traffic light states and digital messages between vehicles or infrastructure elements.

It can function as a well-structured foundation for traffic ontologies such as A.U.T.O.; essentially, it offers the ontology developer a guidance on the relevance of possible roles and concepts. The 6LM provides the additional advantage that it has been unified in an integrated effort between industrial and scientific stakeholders lead by Scholtes [Sch+21] and has been repeatedly applied as a structuring scheme [Sch+24b; Sch+23; Wan+24a].

It is also referenced in the Official Gazette of the German Federal Ministry of Digital and Transport for the systematic description of traffic when applying for an operational domain permit [Dig24], which is required for operating ADSs on public roads in Germany. While the 6LM gives a top-level categorization for the environment, an ontology details and formalizes concepts and roles.

A.U.T.O. is an integrated set of ontologies specifically developed to represent the 6LM as well as various other domains necessary to reason about traffic: For each of the layers described above, there are OWL files containing their respective concepts and roles. Axioms have been incrementally added to the ontology when formalizing a catalog of factors influencing criticality, as described by Westhofen et al. [Wes+22a]. The catalog of factors itself – containing, among others, passing of parking vehicles – was systematically created in a large publicly funded research project involving industry participation. During creation, we have relied on an expert-based brainstorming approach, taking sources such as accident reports and driving school questionnaires into account [Neu+21, Section V A], and also considered possible underlying causal mechanisms [Koo+25]. An example set of A.U.T.O. axioms used in formalizing passing of parking vehicles, given in OWL2 functional-style syntax while ignoring prefixes, is:

```
SubClassOf(Vehicle Dynamical_Object)
EquivalentClasses(Driver ObjectSomeValuesFrom(drives Vehicle))
SubClassOf(
  ObjectIntersectionOf(
    Vehicle
    Non_Moving_Dynamical_Object
    ObjectSomeValuesFrom(
      intersects
      ObjectUnionOf(Parking_Lane Parking_Space)
    )
  )
  Parking_Vehicle
)
EquivalentClasses(
  Non_Moving_Dynamical_Object
  ObjectIntersectionOf(
    Dynamical_Object
    DataHasValue(has_velocity "0.0"^^xsd:double)
  )
)
```

In a nutshell, these axioms formalize that any vehicle is a dynamical object, drivers must drive some vehicle, non-moving vehicles on a parking lane or space as a sufficient condition for parking vehicles, and non-moving as having zero speed. For more examples, refer to the Appendix of the corresponding work by Westhofen et al. (note that temporal properties have since been moved out of the DL axioms into MTCQs) [Wes+22a].

A.U.T.O. additionally detaches concepts semantically spanning over multiple layers of the 6LM. This modularity enables separation of concerns and re-use of concepts, therefore mitigating complexity issues. For example, spatial properties, such as heights or distances,

are prerequisite for entities among almost all layers of the 6LM. Moreover, standardized ontologies for spatial concepts already exist and can therefore be re-used. Here, we opt for the Open Geospatial Consortium's GeoSPARQL ontology, which gives access to DE-9IM and RCC8 vocabularies for representing spatial relations. This enables compatibility of A.U.T.O. to a wide range of available tooling, such as GraphDB[1]. This leads to a lean overall ontology structure, where individual, re-usable domain ontologies are imported, as displayed in Table 5.1. The union of the domain ontologies representing the 6LM and

Table 5.1: Traffic-related ontologies detached from the 6LM for their use in A.U.T.O.

Ontology	Competency	Example Concepts	Example Roles
Geometry	Geometries & shapes	Geometry, Rectangle	Contain, Disjoint
Physics	Physical properties	Dynamical object	Speed, Acceleration
Act	Activities over time	Actor, Event, Activity	Conduct, Participate
Communication	Message exchange	Sender, Receiver	Transmit, Receive
Perception	Perceiving Entities	Observer, Occlusion	Is Occluded For

traffic-related ontologies of Table 5.1 yields the A.U.T.O. Moreover, a Python wrapper around A.U.T.O., called PyAuto [Wes26b], offers an easy creation of scenes, scenarios, placement of traffic participants and road layouts, as well as a basic simulation and visualization framework. This interface is also used for creating the TOBM temporal data and is based on Owlready2 [Lam17].

In practice, we are often confronted with concrete physical data that we require abstracting from. For this, a custom-written library has been integrated into PyAuto, where the user can easily specify definitions for arbitrary concepts and roles using Python code [Wes26a]. For example, defining the concrete semantics of `within` – a role defined in the GeoSPARQL ontology – can be based on Python's Shapely library:

```
from shapely import wkt
with physics:
  class Spatial_Object:
    @augment(AugmentationType.OBJECT_PROPERTY, "within")
    def within(self, other: physics.Spatial_Object):
      return wkt.loads(self.hasGeometry[0].asWKT[0]).
        within(wkt.loads(other.hasGeometry[0].asWKT[0]))
```

The library provides a high level interface enabling users to initiate an augmentation for each individual of a KB.

Data

TOBM includes a benchmark generator to create temporal data for A.U.T.O. with

[1] https://graphwise.ai/components/graphdb

a number of individuals scaling linearly to some $N > 0$. Similar to the two-wheeler benchmark, a seed S is used for pseudo-randomization. From both parameters, it generates scenarios of a certain length (by default, 20 seconds). These can be sampled from two settings:

1. A T-crossing setup with parking vehicles, a pedestrian crossing, bikeway lanes, pedestrians, bicyclists, and passenger cars (cf. Figure 5.2). A bicyclist from the southern arm is specifically programmed to appear behind parking vehicles and subsequently drive over the pedestrian crossing on the western arm without granting right of way. It has $8N + 22$ individuals (including static infrastructure).

2. An X-crossing of two urban roads with traffic signs and dysfunctional traffic lights (cf. Figure 5.3). Compared to the T-crossing, there are only passenger cars and pedestrians but no bicyclists. In total, we obtain $5N + 69$ individuals.

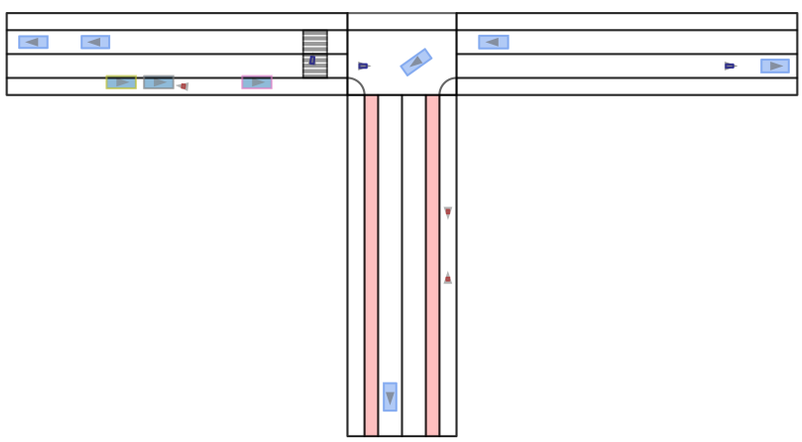

Figure 5.2: A scene of the T-crossing scenario sampled from TOBM for $N = 3$, $S = 0$. Six drivers, three pedestrians, and three bicyclists are present. Three vehicles are parking on the western arm in proximity to a pedestrian walkway. Bicycle lanes are colored red.

The scenarios are created based on behavior models for pedestrians, bicyclists, and passenger cars. The latter two drive up to a speed limit if the area in front is free, otherwise they use a following mode. Vehicles yield on a predicted intersecting path if they are second to reach the intersection point. Moreover, a random successor lane is selected when turning at intersections, giving a turning signal with a probability of 3% each time point. Pedestrians follow their walkway, but can randomly initiate road crossing with a probability of 0.7%. A visualization of an example scenario for each intersection

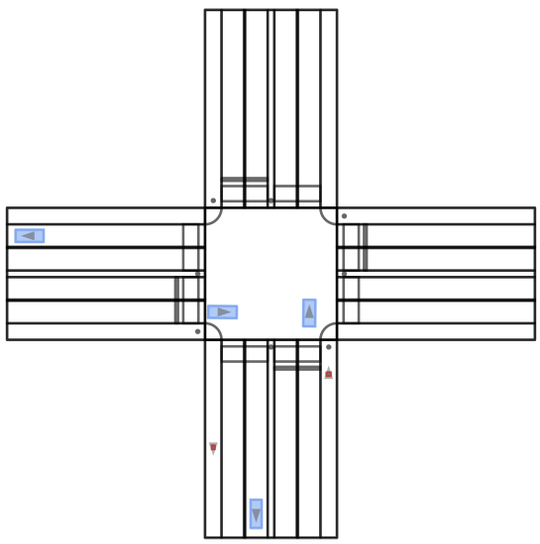

Figure 5.3: A scene of the X-crossing scenario sampled from TOBM for $N = 2$, $S = 0$. Four drivers and two pedestrians are present, as well as traffic lights and pedestrian fords on each arm of the intersection.

can be found in the linked repository of TOBM. Geometrical data are abstracted to spatial predicates, e.g., `within`, using PyAuto.

Figure 5.4 shows the number of data assertions in TOBM for $N \in \{1, \ldots, 5\}$ and $S = 0$ contained in each TKB. For $N = 3$, and 20 seconds sampled with 10 Hertz on the T-crossing setting, this results in a data sequence with 46 individuals and approximately 2 830 facts per time point with constant assertions only counted once, leading to 566 061 assertions over all time points. The largest benchmark, $N = 5$ on the X-crossing, contains 1 027 244 total assertions. Note that the data size is not exactly linear in N, as the probabilistic simulation of additional traffic participants can lead to different spatial configurations. However, as described above, the number of individuals scales linearly with the benchmark size.

Queries

Four queries Φ_1 to Φ_4 are contained in TOBM. Φ_1 has two answer variables and asks for intersecting vulnerable road users for some significant duration.

```
PREFIX l4c: <http://purl.org/auto/l4_core#>
PREFIX l4d: <http://purl.org/auto/l4_de#>
PREFIX phy: <http://purl.org/auto/physics#>

G (l4c:Vehicle(?x) ^ l4d:Vulnerable_Road_User(?v))
```

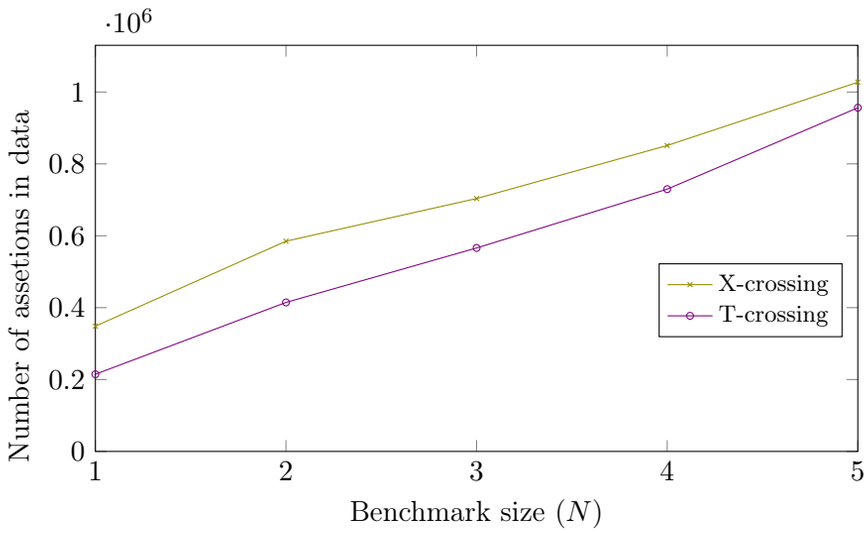

Figure 5.4: Number of total assertions in the TOBM X- and T-crossing data for $N \in \{1, \ldots, N\}$ and $S = 0$, sampled with 10 Hertz and 20 seconds. Static assertions (such as road infrastructure) are counted only once.

```
  &
F
(
  (phy:is_in_proximity(?x,?v)) U_[5,10] (phy:has_intersecting_path(?x,?v))
)
```

Φ_2 (with again two answer variables) addresses the already mentioned passing of parking vehicles on two-lane roads.

```
PREFIX l1c: <http://purl.org/auto/l1_core#>
PREFIX l1d: <http://purl.org/auto/l1_de#>
PREFIX l4d: <http://purl.org/auto/l4_de#>
PREFIX l4c: <http://purl.org/auto/l4_core#>
PREFIX phy: <http://purl.org/auto/physics#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>

G (phy:Moving_Dynamical_Object(?x) & l4c:Vehicle(?x) & l1d:2_Lane_Road(r) & geo:
    sfIntersects(r,?x) & l4d:Parking_Vehicle(?y))
  &
F
(
  (phy:is_in_front_of(?y,?x))
    &
  X[!]
  (
    (phy:is_in_proximity(?x,?y) & phy:is_to_the_side_of(?y,?x))
      U
```

```
    (phy:is_behind(?y,?x))
  )
)
```

Φ_3 formalizes a right turn with two answer variables.

```
PREFIX l1c: <http://purl.org/auto/l1_core#>
PREFIX l1d: <http://purl.org/auto/l1_de#>
PREFIX l4c: <http://purl.org/auto/l4_core#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>

F (l4c:Vehicle(?x) & l1c:Driveable_Lane(l1) & geo:sfIntersects(l1, ?x) & l1d:
    is_lane_right_of(?l2,l1))
  &
F (geo:sfIntersects(?x, ?l2))
```

Φ_4 has three answer variables and concerns a lane change without a signal.

```
PREFIX l1c: <http://purl.org/auto/l1_core#>
PREFIX l1d: <http://purl.org/auto/l1_de#>
PREFIX l4c: <http://purl.org/auto/l4_core#>
PREFIX l6d: <http://purl.org/auto/l6_de#>
PREFIX com: <http://purl.org/auto/communication#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>

G (l4c:Vehicle(?x) & l1c:Driveable_Lane(?l1) & l1c:Driveable_Lane(?l2))
  &
F
(
  (geo:sfWithin(?x,?l1))
    &
  X[!]
  (
    (!(com:delivers_signal(?x,s) & l6d:Left_Turn_Signal(s)))
      U
    (geo:sfIntersects(?x,?l2))
  )
)
```

5.3 Performance Evaluation

Both benchmarks introduced above yields in total 70 possible inputs to Topllet, i.e., combinations of TKBs and MTCQs, on which the performance of both approaches presented in Chapter 4 can now be tested. For this, we pose several questions that we aim to answer by the evaluation.

(Q1) Are the sketched algorithms for MTCQs able to deliver a performance satisfiable for practical applications and how do both compare?

For the automaton-based approach,

(Q2) is the CQ answering optimization for the automaton-based approach described in Section 4.1.4 effective for increasing performance?

(Q3) how much satisfiability knowledge can be gained by the CQ answering run, both locally, i.e., pertaining only to certain time points, and globally, i.e., valid at all times?

(Q4) are there certain answers not found during the second (full) run, or does an under-approximating run suffice in practice?

For the rewriting-based approach,

(Q5) can we find empirical evidence for the theoretically proven statement that it uses UCQ answering more sparingly than the automaton-based algorithm?

(Q6) how much of the performance improvements are due to reducing the dependency on UCQ answering?

Here, practical applicability refers to offline analysis settings as motivated in Chapter 2, where temporal data are queried after collection (in contrast to online monitoring, where results are needed on-the-fly).

5.3.1 Setup and Experimental Design

Three experiments have been conducted. The host system is a Windows 10 machine with an Intel® Core® i9-13900K, 64 GB RAM, and a ten hours time limit per run. The time limit was chosen based on the maximum amount of time that still allowed to execute both algorithms on all benchmarks in a practically feasible duration. Note that determinism of both implementations makes deviations negligible, and therefore a single execution per input combination is deemed sufficient. Code, data, and reproducibility instructions are provided online for each implementation [Wes23a; Wes24].

In the first experiment, we aim to mainly answer Q1 but also collect data for Q5 and Q6. Both Algorithm 1 and Algorithm 3 (with all optimizations enabled) are executed on TOBM $S = 0$ and $N \in \{1, \ldots, 5\}$ as well as the two-wheeler benchmark for $S = 0$ and $N \in \{1, \ldots, 30\}$. To answer the questions posed above, the experimental setup has measured wall clock run times for query answering (called "run time" from here on) for both algorithms. Parsing and loading of queries and TKBs have been excluded, as the goal is to only evaluate the query answering algorithm. Moreover, information on the number of UCQ entailment checks and calls to the CQ answering engine, as well as run time spent in both have been collected.

For answering Q2, we have performed an ablation study where we turn off the CQ optimization of Algorithm 3 in a second experiment, in which we only examine the TOBM T-crossing instance $S = 0$, $N = 3$. Here, we also address Q3 and Q4, for which we additionally measure run times for both the first (under-approximating) and second (full) run when the optimization is enabled. We have also collected the number of answer

candidates for which satisfiability knowledge has been gained at each time point as well as the amount of answers already returned in the first run.

For additional insights into Q5 and Q6, a third experiment on the two-wheeler benchmark for $N \in \{1, \ldots, 30\}$ and a query of the form $\Box\varphi_1 \wedge \Diamond\varphi_2$ allows to empirically assess the relation of superfluous UCQ checks and run time, which we previously motivated from a theoretical perspective using a query of the same form Section 4.3.

5.3.2 Results and Discussion

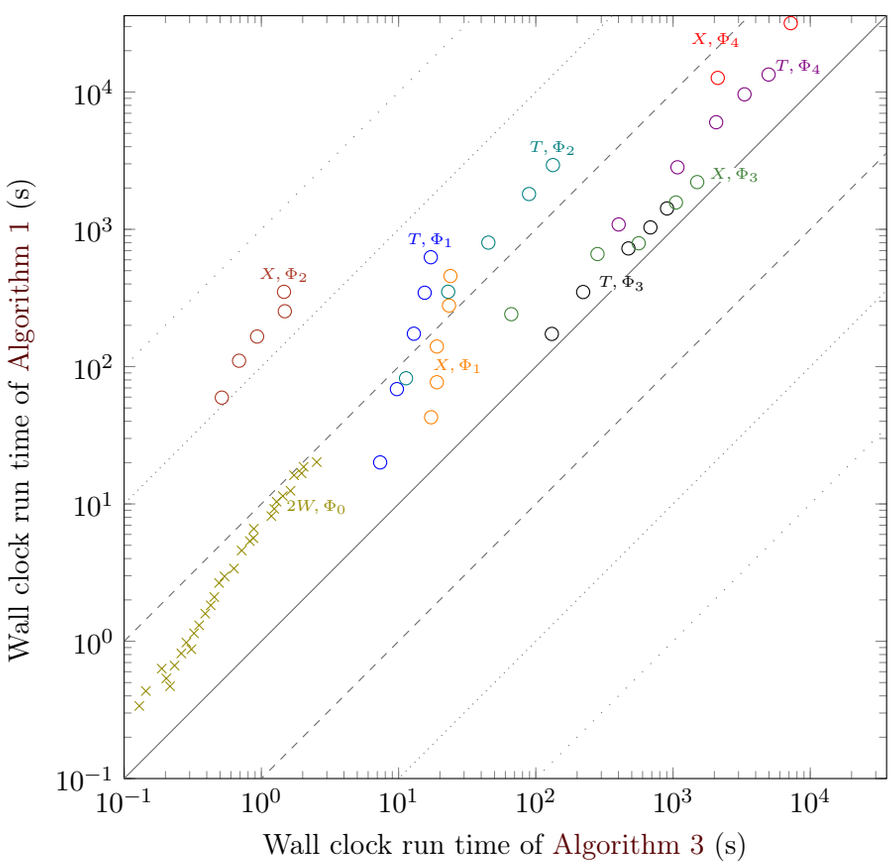

Figure 5.5: Log-scaled scatter plot of the wall clock run times for the automaton-based Algorithm 1 against the rewriting-based Algorithm 3 on TOBM (X and T, circles) and the two-wheeler benchmark (2W, crosses). Colors indicate the membership of scaled benchmarks to the same type. Dashed lines represent orders of magnitude.

Q1 The run times measured in the first experiment are depicted in Figure 5.5. Memory exceedance has not been observed on the benchmark inputs. Already the automaton-based

approach offers an adequate performance: For the average benchmark size of $N = 3$ on the TOBM T-crossing, run times were 2.9 minutes (Φ_1), 13.3 minutes (Φ_2), 12.1 minutes (Φ_3), and 100.4 minutes (Φ_4). The rewriting algorithm beats these numbers significantly on all benchmarks. On the mentioned examples, it requires instead only 0.2 minutes, 0.8 minutes, 7.9 minutes, and 34.4 minutes. We achieve an average speedup of 19.09 over all benchmarks, that is, over an order of magnitude, with a standard deviation of 45.79. Hence, the rewriting-based approach offers satisfiable performance on realistic input sizes sufficient for *offline* data analysis, considering that the experiments were performed on consumer-grade desktop hardware. It has to be remarked that all queries contained at most four answer variables; it is likely that a greater number impacts performance negatively – evidence for that is Φ_4, which has three answer variables and requires more run time compared to the other TOBM queries containing only two variables.

The performance does not suffice for online monitoring. Only small data and ontologies, such as the two-wheeler benchmark, can be handled by the engines under sufficient time constraints. For $N = 15$ on the two-wheeler benchmark (which contains 45 assertions per time point, on average), the rewriting algorithm required 164 ms per time point. This allows for an input frequency of roughly six Hertz, which can suffice in some online settings.

Run times do not always scale strictly exponential with input size for both algorithms, as, e.g., evident for Φ_2 and Φ_4 on the T-crossing. This is due to relying on efficient, i.e., in practice often non-exponential, CQ answering for a significant portion of the search space. Especially for the automaton-based algorithm, however, some queries induce, in fact, an exponential relationship. An example is Φ_1 on the X-crossing, where the time spent on CQ answering is constant over the benchmark size.

A timeout was observed on three benchmarks on the FA-based implementation (Φ_4 on the X-crossing for $N \geq 3$). We will see later that many candidates could be excluded on these benchmarks by the CQ optimization; however, the remaining checks still put a heavy burden on the UCQ engine. For these three timeouts, with rewriting, we can answer the queries in 3.52, 5.72, and 7.88 hours.

Q2 We now compare the run time when enabling Algorithm 2 to the basic version of Algorithm 1 without the CQ answering optimization in Figure 5.6. Wall clock times of both the CQ answering run t_1 (“first run”) and the full-semantics run t_2 (“second run”) are also depicted. The results show that the naïve automaton-based approach does not offer adequate performance on the TOBM T-crossing for $N = 3$, even for queries with two answer variables. On average, run times are increased by a factor of 111.3, i.e., two orders of magnitudes.

In the optimized version, a significant duration is still spent using the expensive, full semantics check despite iterating only through a fraction of all candidates (cf. Table 5.2). Therefore, the high run times of the naïve algorithm do not surprise; it appears that the

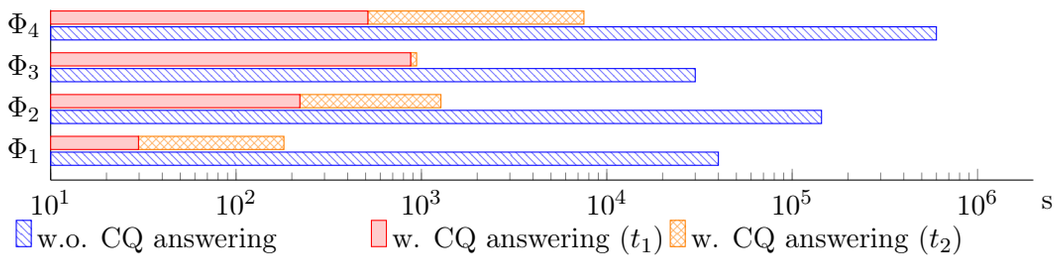

Figure 5.6: Log-scaled run times of Algorithm 1 with and without the CQ answering optimization enabled for the TOBM T-crossing $S = 0, N = 3$. Run times without the optimization are extrapolated after one hour. The times spent in the first (t_1) and second run (t_2) are additionally given for the optimized implementation.

UCQ engine is vastly less efficient than its CQ counterpart. Hence, leveraging the CQ engine makes MTCQ answering practically feasible. However, some queries may trigger special cases in the optimizations of the CQ engine, leading to higher run times, e.g., role inclusion axioms for Φ_3 where already the CQ engine requires 14.6 minutes.

Q3 and Q4 The canonical question regards where the performance improvements of the optimized version are generated, as the effect of using CQ answering can be twofold: Firstly, a set of candidates can be excluded globally: for a candidate, the CQ checks sufficed to either arrive at a final state after or show that no final state is reachable. Secondly, even if a candidate was not globally excluded, it generates "local" (non-)answers that can be cached for subsequent checks of Point 1 of Lemma 4.1.1. We thus report both exclusions, averaged over all time points and checked edges at each time point, in Table 5.2 for the same benchmark used for Q2. Q4 asks whether the second run is actually worthwhile. Table 5.2 reports how many certain answers (called $\mathsf{cert}_{\mathcal{K}}$) were already found in the first run (called $\mathsf{cert}^1_{\mathcal{K}}$).

Table 5.2: Effects of the CQ answering optimization on Algorithm 1 for the TOBM T-crossing $S = 0, N = 3$.

Query	Φ_1	Φ_2	Φ_3	Φ_4
Globally excluded candidates (%)	97.88	99.29	97.88	99.71
Globally and locally excluded candidates (%)	98.73	99.55	99.54	99.80
$\lvert\mathsf{cert}^1_{\mathcal{K}}\rvert / \lvert\mathsf{cert}_{\mathcal{K}}\rvert$	1	1	1	1

The results show CQ answering to aid mainly by excluding candidates globally. In practice, few combinations are entailed, and hence most of these candidates violate the

global type constraints, which leads to being stuck in an accepting sink (and therefore, the candidate will be rejected overall). Local exclusion has minor but non-negligible effects, e.g., avoiding on average 42 additional candidates for Φ_3. Moreover, all certain answers were already found in the first run, indicating suitability of using solely the incomplete first run if no formal guarantees are required.

Leveraging CQ answering has its limitations. For Φ_4 on the X-crossing and $N = 2$, the first run excluded the majority of the candidates, but performance was still suboptimal. Specifically, exclusion occurred to 99.83% of candidates, which took 2.38 minutes, leaving 960 candidates for the second run. As alluded to when discussing the timeouts of Figure 5.5, this is no small task: for 200 time points in the data this leaves 180 seconds per time point to finish within the time limit of 10 hours. Hence, each candidate must not take up more than 0.1875 seconds per time point on average, which requires checking multiple edges in multiple states. Experiments indicate each edge check to take a two-digit millisecond duration. Thus, the automaton-based implementation is not able to handle large candidate sets in the second run of the automaton-based approach efficiently. A possible mitigation for the UCQ engine involves using approaches similar to what has been done for CQ answering (such as binary instance retrieval). However, it is not directly clear how to do so as disjunctions cannot be rolled-up or examined "atom-by-atom."

Q5 and Q6 For Q1, we showed that the rewriting-based approach has improved run times on all benchmarks. The theoretical analysis of Section 4.3 hinted that one underlying reason might be the more effective use of CQ and UCQ answering. We now evaluate this hypothesis by assessing the calls to both non-temporal query engines and the time spent there, which allows us to attribute time savings to more efficient use of UCQs. Table 5.3 and Table 5.4 show the number of UCQ entailment checks, calls to the CQ engine, and their run times on the largest data of the TOBM X- and T-crossing settings as well as the two-wheeler benchmark.

In the data, UCQ answering takes up most of the time in the automaton-based approach, indicating that decreasing the number of UCQ entailment checks promises large benefits. In four cases, the rewriting-based approach eliminated superfluous UCQ checks altogether, leading to improved overall performance. In general, even a few UCQ entailment checks entail high run times, which can be explained by UCQ answering being less optimizable in comparison to CQ engines.

For the final question, Q6, we are interested in how much of the improvements are due to more efficient UCQ answering, which is presented in Table 5.5 for the largest data of each benchmark. There is an overall 86% reduction in run time, of which, on average, 70% is due to better UCQ handling. On six of the nine benchmarks, run times are improved largely, i.e., over 80%, solely by saving time spent answering UCQs. Three exceptions exist in Table 5.5: For Φ_2 on the TOBM X-crossing, no reduction is present as already the automaton-based implementation did not require UCQ checks. Second, Φ_3 on

Table 5.3: Number of UCQ entailment and CQ answering calls, their run times, and overall run times (in seconds) for Algorithm 1 on the largest benchmark of each benchmark family where no timeout occurred.

TKB	Query	UCQs	t_{UCQ} (s)	CQs	t_{CQ} (s)	t (s)
TOBM T-Crossing	Φ_1	94 720	590.61s	593	35.70s	626.31s
	Φ_2	59 061	2 407.80s	1 203	530.42s	2 938.23s
	Φ_3	17 299	115.29s	603	1 304.76s	1 420.05s
	Φ_4	962 613	12 812.79s	1 203	595.79s	13 408.60s
TOBM X -Crossing	Φ_1	27 824	392.27s	593	64.74s	457.02s
	Φ_2	0	0s	1 203	350.72s	350.73s
	Φ_3	16 780	329.33s	603	1 879.35s	2 208.70s
	Φ_4	1 697 010	31 606.45s	1 203	126.68s	31 733.18s
2-Wheeler	Φ_0	5 232	17.51s	45	2.66s	20.19s

Table 5.4: Number of UCQ entailment and CQ answering calls, their run times, and overall run times (in seconds) for Algorithm 3 on the largest benchmark of each benchmark family where no timeout occurred.

TKB	Query	UCQs	t_{UCQ} (s)	CQs	t_{CQ} (s)	t (s)
TOBM T-Crossing	Φ_1	0	0s	397	17.03s	17.17s
	Φ_2	7 037	92.42s	602	40.58s	133.23s
	Φ_3	0	0s	402	904.00s	904.20s
	Φ_4	136 732	4 793.19s	602	177.03s	4 970.80s
TOBM X-Crossing	Φ_1	0	0s	397	23.77s	23.88s
	Φ_2	0	0s	1	1.20s	1.46s
	Φ_3	0	0s	402	1 500.09s	1 500.33s
	Φ_4	204 800	7 188.71s	602	17.58s	7 206.80s
2-Wheeler	Φ_0	357	0.27s	8	2.21s	2.53s

both X- and T-crossing, the CQ engine takes up most of the time (roughly 85% and 92%, respectively) in Algorithm 1. Reducing UCQ answering has limited potential in overall performance improvements in these cases. Still, Algorithm 3 offers a significant speedup on those benchmarks as well by decreasing the time spent on CQ answering. Recall that the initial run checks various CQs in hope to extract some satisfiability information – not all of these checks are strictly required for answering the query. The experimental data show that rewriting-based approach seems to use CQs more sparingly as well.

Table 5.5: The percentage of the overall saved run time of Algorithm 3 that is due to savings in UCQ entailment checks compared to Algorithm 1, measured on the largest data of each benchmark where no timeout occurred.

TKB	Query	Run Time Saved (s)	Saved by UCQs (%)
TOBM T-Crossing	Φ_1	609.14	96.96
	Φ_2	2 805.00	82.55
	Φ_3	515.85	22.35
	Φ_4	8 437.80	95.04
TOBM X-Crossing	Φ_1	433.14	90.56
	Φ_2	349.27	0.00
	Φ_3	708.37	46.49
	Φ_4	24 526.38	99.56
2-Wheeler	Φ_0	17.66	97.62

Section 4.3 gave a class of queries where UCQs are not required on any TKB at all – namely $\mathrm{TCQ}_{\mathrm{CQ}^*}$. Of the queries introduced above, only Φ_3 is in $\mathrm{TCQ}_{\mathrm{CQ}^*}$, on which no UCQ checks were performed by the automaton-based approach. Since UCQ checks were reduced on all other queries as well, it can be deduced that the rewriting-based algorithm has advantages also when being applied outside $\mathrm{TCQ}_{\mathrm{CQ}^*}$.

It remains to empirically analyze the behavior of both algorithms in $\mathrm{TCQ}_{\mathrm{CQ}^*}$. For this, we use the query

$$\Box\texttt{TwoWheeledVehicle}(x) \wedge \Diamond\texttt{System}(x) \in \mathrm{TCQ}_{\mathrm{CQ}^*},$$

for which Section 4.3 showed that there is a TKB where the automaton-based approach uses UCQs. In contrast, Algorithm 3 solves it solely by CQ answering. The query was executed on the two-wheeler benchmark while measuring the number of UCQ checks. The results are shown in Figure 5.7. We confirm the theoretical analysis that Algorithm 3 does not require UCQs, whereas Algorithm 1 must, in fact, check disjunctive queries. As the query has only one variable and UCQ checks increase linear with the size of the benchmark (proportional to the number of two-wheeled vehicles), the observed linear run time of Algorithm 1 is to be expected. Moreover, Figure 5.7 highlights the effectiveness of the employed CQ engine: The performance is essentially constant, despite the fact that $N = 30$ contains over six times the assertions of $N = 1$. A likely explanation is that the answers can already be returned by information gathered from the completion graph generated during the initial consistency check, which in turn can be done efficiently due to the small number of axioms in the ontology.

A noteworthy caveat of the analyses is their specificity to the employed non-temporal

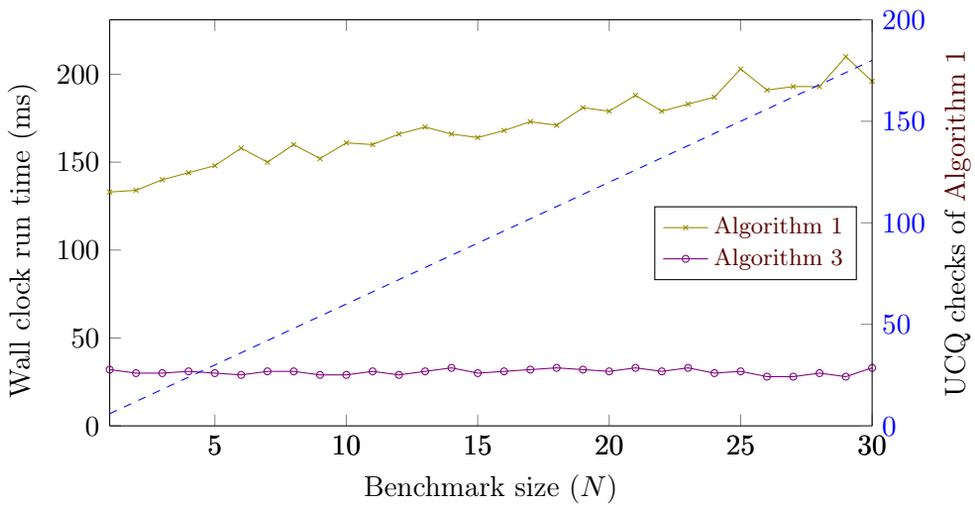

Figure 5.7: Wall clock run times of the rewriting-based Algorithm 3 vs. the automaton-based Algorithm 1 and its number of UCQ entailment checks for the two-wheeler benchmark on the MTCQ $\Box\texttt{TwoWheeledVehicle}(x) \wedge \Diamond\texttt{System}(x) \in \mathrm{TCQ}_{\mathrm{CQ}^*}$. Algorithm 3 uses no UCQ checks.

query engines on which both implementations rely. Currently, no other UCQ engines for expressive DLs are known. Hence, the experiments have been restricted to what has been implemented in Topllet. In general, the performance improvements attributable to better UCQ handling might be lower for more efficient UCQ engines, or even higher for those that are less optimized. However, since UCQ answering is practically more challenging than CQ answering, it can be conjectured that the number of calls to a query engine, as assessed for Q5 and Q6, is a reasonably generalizable proxy.

5.4 Integration into Scenario Coverage

The application context of this dissertation is assessing SOTIF achievement by using collected field data, as described in Chapter 2. For this, it is first required to assess the quality of said data. We are specifically interested in checking whether we have *enough* data for the identified causal chains preceding a harm. Recall the example in Figure 2.2: During virtual and real-world test drives, it must be ensured that the data cover all triggering conditions (ice plates existing and falling from truck), the subsequent hazardous behaviors and hazards (concerning distance, lane keeping, and lane changing behavior), as well as additional scenario conditions (such as ice on road and vehicle being on other lanes). In what follows, these conditions are referred to as *scenario classes*, which is simply a constraint representing a set of concrete scenarios.

Coverage of scenario classes by recorded data aims at quantifying the quality of the matching between the data records and specified scenario constraints. Historically, coverage has been well-established in software engineering, including concepts like branch, statement, and MC/DC coverage [ZHM97]. In automated driving, covering all relevant scenarios with data currently forms one of the fundamental safety challenges [Neu+20; Rie+20]. In Germany, proving adequate coverage is even required for approval, as per the applicable regulation: "Test cases must provide sufficient test coverage for all scenarios, test parameters, and environmental influences. The coverage must be justified [...]" [Ver22b]. It can be argued that coverage should not be blindly followed as the sole indicator in software engineering [HT90; Gay+14]. However, for complex, automated systems, if metrics are of high quality, it can be a valuable asset in the practitioner's toolbox [AHR15], especially for guiding data collection efforts.

The most straightforward way of implementing coverage for automated driving applications has been coined "tag-based coverage" [GBO24; Sch+24b], which is simply the proportion of scenario classes for which an instance was observed in the data. More involved coverage metrics exist in our assessed domain [Gel+25, Section 3.3]; among these are

- tree-based coverage, which removes implausible combinations by constraints along a tree structure [Sch+24b],
- time-based coverage, which measures the recorded duration for which no scenario class is applicable [GBO24],
- probabilistically-bounded coverage, by counting only combinations exhibiting a certain likelihood [AHR15],
- t-wise coverage, with the assumption that many harms are triggered by at most t scenario classes, and hence considering only t-combinations [Sun+22],
- quantitative k-projection coverage, which extends t-wise coverage by weights and rules [CHY18], and
- concrete scenario coverage, where each of the (finitely many) concrete scenarios forms a unique scenario class [AW19a].

For reasons of simplification, we use tag-based coverage as a canonical foundation for showing how MTCQs and hence the OWA can be incorporated into coverage frameworks. By this, we put MTCQs into a larger context and highlight the implications of transitioning from a CWA to an OWA in automated driving applications.

Evaluating any of the aforementioned metrics requires recognizing scenario classes in data, for which temporal logics are a natural choice. These include MTL, STL, LTL, and CMFTBL, as presented in Chapter 2. All these approaches make the CWA.

This has several downsides: First, the CWA assumes a perfect ground truth. However, data incompleteness stemming from perception imperfections (think: occlusion or false calibration) can occur. Second, classification under CWA settings requires a complete specification to be available. Complexities of the Operational Design Domain (ODD), such as a wide array of possible objects to encounter, and vagueness of their definitions, e.g., due to fuzzy legal texts, can invalidate this assumption.

A possible solution to handling incomplete data and specifications is the OWA, as already introduced in Chapter 2. The OWA allows incorporating absence of knowledge into coverage by inferring no additional facts from missing information. After classification, observed scenarios can either belong to exactly one scenario class, or, due to the OWA, the actual class could not be determined exactly and hence multiple possible classes must be considered. Take as an example a recorded scenario in which rain is present, where sensor imperfections allow determining neither the presence nor absence of a lead vehicle. It is thus not clear if the hazardous behavior "safety distance violated" is given.

This example makes clear that real-world coverage computation must regard data records for which the actual scenario class is unknown. Currently, no formalism for incorporating ambiguity into scenario coverage exists. Based on what has been proposed by Westhofen et al. [Wes+26], this section advances the state-of-the-art by

1. extending the standard coverage problem with an OWA,
2. defining corresponding problems for lower and upper bounds on coverage within this novel framework,
3. analyzing complexity of these problems and giving tractable algorithms for computing both bounds, and
4. showing that adequate computations of OWA coverage with Topllet are possible using synthetic and real-world data.

In addition to what is presented in the original work [Wes+26], this dissertation extends the experiments on real world data from one to four MTCQs while also reporting run time of Topllet, enabling more comprehensive conclusions about the applicability of the tool.

5.4.1 Formal Framework

As motivated above, we assume finitely many data traces $(d_1, \ldots, d_n)$ as a result of field data collection. Each data trace d_i can be simply represented as a TKB, i.e., $d_i = (\mathcal{O}, (\mathcal{D}_j)_{j \in \{0,\ldots,m\}})$ for an ontology $\mathcal{O}$ and nontemporal databases $\mathcal{D}_j$. The set $\mathbb{D}$ denotes all possible nontemporal databases. As we adopt tag-based coverage, we assume to have a finite set of tags T and finitely many possible values V_t for each tag $t \in T$. The

set of all values is $V = \bigcup_{t \in T} V_T$. We can now represent the data simply in terms of its tag assignments, i.e., for each data trace d_i we assume that there is a scenario represented as a function $s_i : T \to V$ that assigns each tag its value (depending on what has been observed in the data). Converting data to scenarios can be implemented as a function from all data traces to tag evaluations: $\Lambda_{T,V} : \bigcup_{h>0} \mathbb{D}^h \to V^T$ for tags T and values V, with $V^T = \{f : T \to V\}$. We will later see how MTCQs can be used for defining $\Lambda_{T,V}$, but leave the subsequent discussion on coverage deliberately independent of MTCQs. The translation via $\Lambda_{T,V}$ finally yields a list of scenarios $(s_1, \ldots, s_n)$ from the data traces $(d_1, \ldots, d_n)$.

Example 5.4.1

Assume four tags $T = \{t_1, \ldots, t_4\}$ with Boolean values $V = \{0, 1\}$ (denoting absence and presence of the tag, respectively). We use examples of potential triggering conditions and hazardous behaviors for the definition of the example tags:

t_1: Proximity to a bicycle (which possibly triggers hazardous behavior of the ADS due to difficult-to-predict movements).

t_2: Passing of parking vehicles (a potential triggering condition due to opening doors and occlusion of other traffic participants).

t_3: Violation of the safety distance (a hazardous behavior by the ADS).

t_4: Bicycles driving from a walkway onto the road surface in proximity to the system (a triggering condition in case such behavior is not anticipated correctly or timely).

Then, the scenario $s_1 : t_1 \mapsto 0, t_2 \mapsto 0, t_3 \mapsto 0, t_4 \mapsto 1$ encodes the ADS encountering a walkway-road-changing bicycle without any of the three other conditions being present.

Tag-based coverage, in its standard CWA definition [Sch+24b], is the ratio between observed and all possible scenario classes for scenarios $(s_1, \ldots, s_n)$:

$$\kappa(\mathbb{T}, V, s_1, \ldots, s_n) = \frac{|\{s_i \mid i \in \{1, \ldots, n\}\}|}{|\mathbb{T}|}. \tag{5.1}$$

Here, $\mathbb{T}$ denotes the set of all possible scenarios. While the size of $\mathbb{T}$ can be computed based on dependencies between tags [Sch+24b], the simplest case is $\mathbb{T} = V^T$, which is also assumed from here on.

When advancing coverage from a CWA to the OWA, it can occur that the value of some tags might be unknown for a given data trace. As motivated above, this can be caused by specification fuzziness or perception issues. We model this behavior by allowing for a unique symbol $*$ in the values V_t of any tag t that shall be interpreted under the OWA. Note that not all tags require an OWA, and we hence still allow some tag values

to not contain $*$. Then, $s_i(t) = *$ implies that $s_i(t)$ is ambiguous and any value from $V_t \setminus \{*\}$ is applicable. Only one value could have actually been valid. Hence, a set of possible options for a scenario s_i opens up:

$$\mathsf{Options}(s_i) = \left\{ o : T \to V \;\middle|\; \begin{array}{c} \forall t \in T.\ s_i(t) \neq * \to o(t) = s_i(t) \ \wedge \\ s_i(t) = * \to o(t) \in V_t \setminus \{*\} \end{array} \right\}.$$

In the worst case, this creates an exponential set of entries for each scenario ($|V|^{|T|}$ many to be precise). This implies that the subsequently examined problems are exponential in the number of tags. Since collected field data can grow excessively over the course of testing and operation, we are specifically interested in the dependency of coverage on the data. This is comparable to what has been proposed as data complexity [Var82]. We therefore operate directly on $\mathsf{Options}(s_i)$ from here on and formulate all problems in terms of $S = (S_1, \ldots, S_n) = (\mathsf{Options}(s_1), \ldots, \mathsf{Options}(s_n))$ as inputs. This prevents the problems from being trivially exponential and allows us to identify subtleties in complexity later on.

Example 5.4.2

Take the first tag: Proximity to a bicycle. In case of an extremely high speed of the perceived vehicle, e.g., 12 meters per second, it must be questioned whether the given information is correct. It might well be a motorcycle, however, it is surely possible to achieve such speeds without any motorization. At such speeds, the tag value can be considered unclear. Due to this perception impreciseness, we interpret the tag under an OWA.

An OWA can be made for the other tags as well: For t_2, specification vagueness arises: it is not always clear how "parking vehicles" are defined. For example, we can surely say that any non-moving vehicle on a parking spot is parking; contrarily, any non-moving vehicle cannot be parking. A gray zone arises for non-moving vehicles that are not within parking spots (which could obviously be narrowed further). For illustration purposes, we use the rather naïve definition above.

The third tag is afflicted with (intentional) legal fuzziness in the definition of safety distance. Note that an exact safety distance depends on many, often unknown factors of both the leading and following vehicle, such as tire and brake temperatures, the coefficient of friction between road and tires, and the vehicle's mass. As a simplifying assumption, engineers might first consider the well-known "half-speedometer" rule as a safe over-approximation of the actual safety distance (as also indicated by its use in German jurisdiction). For the scope of this example, the limits of the UN regulation 157 act as an under-approximation [Uni21]. The resulting gray zone is depicted in Figure 5.8.

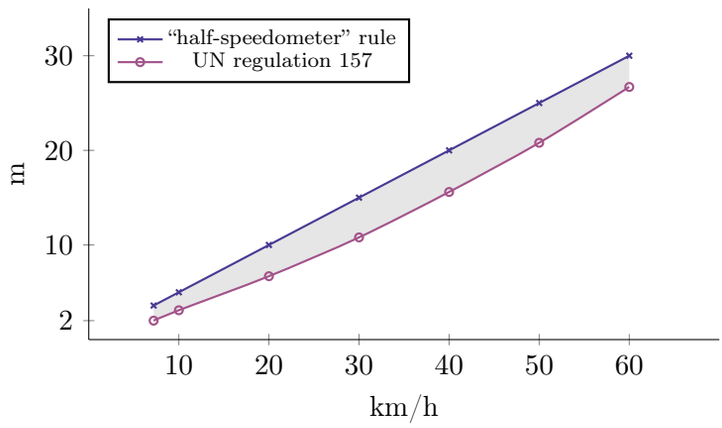

Figure 5.8: Safety distances according to the "half-speedometer" rule and the UN regulation 157, with the gray zone highlighted in between.

The final tag, t_4, can be interpreted under an OWA due to perception uncertainties, similar to t_1: Bicycles and road infrastructure might not be correctly identified. We conservatively correct for this by including, for the positive tag value, only bicycles with speeds between two and seven meters per second. If a bicycle is too slow, it might be the case that the driver dismounted. In case of high-speed bicycles on walkways, the perception results can be questioned. We certainly exclude bicycles that are either never in proximity or always behind the system. This leaves a gray zone for those bicycles that are eventually close to the front of the vehicle but the transition from a walkway could not be observed, as well as having a too low or high speed.

Formally, we add an OWA by extending the values of all tags $i \in \{1, \ldots, 4\}$, V_{t_i}, from $\{0, 1\}$ to $\{0, 1, *\}$. Here, $*$ means that the tag value could not be determined. For a given scenario s with $s(t_i) = *$ for $i \in \{1, \ldots, 4\}$ (no tag value was determinable), 16 options arise., i.e., $\mathsf{Options}(s) = \{o_1, \ldots, o_{16}\}$. For instance, the first option is to set any tag value to zero: $o_1(t_i) = 0$ for all $i \in \{1, \ldots, 4\}$.

5.4.2 Definition of Lower and Upper Coverage Bounds

If we assume the OWA for a list of scenario options $(S_1, \ldots, S_n)$, we cannot retrospectively compute the exact coverage value – as there are many possibilities. We must therefore rely on identifying lower and upper bounds on the generally indeterminable κ:

$$\frac{\kappa_{min}}{|\mathbb{T}|} \leq \kappa(\mathbb{T}, (s'_1, \ldots, s'_n)) \leq \frac{\kappa_{max}}{|\mathbb{T}|}.$$

Here, $s'_i \in S_i$ denotes the "actual" i-th scenario. Intuitively, to determine the values of both κ_{min} and κ_{max}, we must select an element from each S_i s.t. the number of overall selected elements is minimal and maximal, respectively. This is formalized in the following two problems representing the lower and upper bound.

Definition 5.4.1 (MinCover)

For a list of non-empty finite sets $S = (S_1, \ldots, S_n)$ over a universe $U = \bigcup_{i \in \{1,\ldots,n\}} S_i$, find κ_{min} which is the smallest $k \leq n$ s.t. there is a function π with $|\mathrm{Im}(\pi)| = k$ assigning each index i an element of S_i, i.e., $\pi : \{1, \ldots, n\} \to U$ where $\pi(i) \in S_i$ for all $i \in \{1, \ldots, n\}$.

Definition 5.4.2 (MaxCover)

For a list of non-empty finite sets $S = (S_1, \ldots, S_n)$ over a universe $U = \bigcup_{i \in \{1,\ldots,n\}} S_i$, find κ_{max} which is the largest $k \leq n$ s.t. there is a function π with $|\mathrm{Im}(\pi)| = k$ assigning each index i an element of S_i, i.e., $\pi : \{1, \ldots, n\} \to U$ where $\pi(i) \in S_i$ for all $i \in \{1, \ldots, n\}$.

We call the corresponding decision problems IsMinCover and IsMaxCover. They ask whether some given $k \in \mathbb{N}$ equals κ_{min} and κ_{max}, respectively.

Example 5.4.3

Take again the tags from Example 5.4.1. We will use $(s_i(t_1), \ldots, s_i(t_4))$ to concisely denote the function s_i, e.g., $(1, 0, 0, 1)$. For better presentation, we also refer to the scenario as its induced list of tags with value 1, e.g., (t_1, t_4). We are presented with four scenarios (s_1, s_2, s_3, s_4) extracted from the recorded data: $s_1 = (0, 0, 0, 1)$, $s_2 = (1, 0, 0, 1)$, $s_3 = (*, 0, 0, 1)$, and $s_4 = (*, *, 0, 1)$.

For the first two data points, the bounds κ_{min} and κ_{max} coincide with κ due to missing ambiguity. In the third record, the first tag (proximity to a bicycle) was indeterminable, but this does not change coverage as any possible option has already been observed before. Finally, the fourth scenario (where also passing of parking vehicles is unknown) leads to a divergence of the lower and upper bound. The possible options for all scenarios are:

$$\mathsf{Options}(s_1) = \{s_1\} = \{(0, 0, 0, 1)\} = \{(t_4)\}$$
$$\mathsf{Options}(s_2) = \{s_2\} = \{(1, 0, 0, 1)\} = \{(t_1, t_4)\}$$
$$\mathsf{Options}(s_3) = \left\{ \begin{matrix} (\mathbf{0}, 0, 0, 1) = (t_4), \\ (\mathbf{1}, 0, 0, 1) = (t_1, t_4) \end{matrix} \right\}$$
$$\mathsf{Options}(s_4) = \left\{ \begin{matrix} (\mathbf{0}, \mathbf{0}, 0, 1) = (t_4), \\ (\mathbf{0}, \mathbf{1}, 0, 1) = (t_2, t_4), \\ (\mathbf{1}, \mathbf{0}, 0, 1) = (t_1, t_4), \\ (\mathbf{1}, \mathbf{1}, 0, 1) = (t_1, t_2, t_4) \end{matrix} \right\}$$

Values originating from unknown entries are bold-faced. Based on the enumeration of all options, we can determine the worst and best case assignment for coverage: For the

first, we can simply assign s_3 and s_4 one of s_1 or s_2 (this minimizes coverage to two unique scenarios). The best case is to select a novel scenario for s_4, but, in any case, $s_1, s_2 \in \mathsf{Options}(s_3)$ and hence coverage cannot reach four unique scenarios.

5.4.3 Computing the Lower and Upper Coverage Bound

The naïve option for computing both bounds is to simply compute coverage for all possible combination of options, which has an exponential dependency on n, the number of data traces. Preliminary experiments show that this brute-force algorithm is infeasible already on small datasets. We specified seven tags. Three of them are interpreted under the OWA, where for the OWA tags a 10% chance leads to a $*$-entry. For both CWA and OWA tags, a fair coin flip is performed to get the Boolean value of the tag (if required, i.e., if the entry is not already chosen to be $*$). For this setup, the brute-force algorithm did not terminate within one hour on 10^2 scenarios on a standard consumer notebook.

Intuitively, one might be tempted to formulate the problem as input to highly optimized SMT solvers instead. For encoding the bound computation in SMT, we represent the i-th scenario S_i by introducing a variable t_i for each tag $t \in T$ and define the constraint

$$\zeta_i = \bigwedge_{t \in T} t_i \in S_i(t),$$

enforcing the observations of the i-th scenario, i.e., t_i can be either a constant value (if not $*$) or take any of the tag's values (if $*$). We can then encode uniqueness of the i-th scenario with an if-then-else (ite) operator as

$$\iota_i = \text{ite}(\bigwedge_{j \in \{1,\ldots,i\}} \bigvee_{t \in T} t_i \neq t_j, 1, 0),$$

i.e., ι_i is 1 if the i-th scenario is unique to all previous scenarios, and 0 otherwise. Assume we have computed the lower and upper bounds for iteration $i-1$ as k^{i-1}_{max} and k^{i-1}_{min} (starting with 0 for the first iteration). To check whether we can stay at the lower bound, i.e., $k^i_{min} = k^{i-1}_{min}$, we check for satisfiability of

$$\bigwedge_{i \in \{1,\ldots,n\}} \zeta_i \wedge \sum_{i \in \{1,\ldots,n\}} \iota_i = k^{i-1}_{min}.$$

Similarly, for testing whether to increase the upper bound, i.e., $k^i_{min} = k^{i-1}_{min} + 1$, we can check

$$\bigwedge_{i \in \{1,\ldots,n\}} \zeta_i \wedge \sum_{i \in \{1,\ldots,n\}} \iota_i = k^{i-1}_{max} + 1.$$

This iterative approach is able to re-use information from previous runs, which is especially useful when data are gathered incrementally by continuous field tests. Additional practical

optimizations can be implemented, such as not running the solver if no $*$ are present in the given scenario.

As SMT solvers are highly optimized, this technique performs better than the brute-force algorithm. Yet, SMT solving is still an NP-hard problem. Our experiments with the Z3 solver [MB08] on the same setup mentioned above show non-termination of this approach within one hour already for 10^3 scenarios, even if bounds from the prior iteration are reused.

Thorough theoretical examinations of both problems are thus in order. As we will see, IsMinCover is not efficiently solvable: it is D^p-complete (recall that any problem in D^p can be represented as the intersection of an NP- and co-NP-problem). However, there is a practically tractable problem in which MinCover can be encoded. This enables computing the lower bound on larger data sizes. IsMaxCover, on the other hand, is in P. Theoretically, both brute-force and SMT are therefore suboptimal for the upper bound.

MinCover

As stated above, IsMinCover is a D^p-complete problem. Membership in D^p can be intuitively derived from the observation that it requires checking whether there is a function with the mentioned constraints (an NP problem) and checking whether there is no such smaller function (a coNP problem). Hardness can be shown by a reduction from SAT-UNSAT, which is a standard D^p-complete problem [PY82]. For this, we will use a set-based notion of Boolean CNF formula as follows: Let $X = \{x_1, \ldots, x_u\}$ be a set of variables and $\overline{X} = \{\neg x \mid x \in X\}$ their negations. A CNF, in the context of this section, is viewed a set of clauses $C = \{C_1, \ldots, C_v\}$ where each clause is a finite set of literals from $X \cup \overline{X}$. A solution to a CNF C is a function that satisfies all of its clauses, i.e., $\alpha : X \to \{0, 1\}$ where for all $C_j \in C$ there is some $l \in C_j$ s.t. $\alpha(l) = 1$ if $l \in X$ or $\alpha(x) = 0$ if $l \in \overline{X}$.

Theorem 5.4.1 ([Wes+26])

IsMinCover is D^p-complete.

Proof 5.4.1 (of Theorem 5.4.1 [Wes+26])

IsMinCover is the intersection of an NP- and coNP-problem: IsMinCover $= \{L_1 \cap L_2 \mid L_1 \in \text{NP}, L_2 \in \text{coNP}\}$ with

$$L_1 = \big\{\big((S_1, \ldots, S_n), k\big) \mid \text{exists } \pi : \{1, \ldots, n\} \to U \text{ s.t. } |\text{Im}(\pi)| = k \text{ and } \pi(i) \in S_i \text{ for } i \in \{1, \ldots, n\}\big\},$$

which can be decided by polynomially verifying both constraints given a certificate function π and

$$L_2 = \big\{\big((S_1,\dots,S_n),k\big) \mid \text{not exists } \pi : \{1,\dots,n\} \to U \text{ s.t. } |\mathrm{Im}(\pi)| < k \text{ and } \pi(i) \in S_i \text{ for } i \in \{1,\dots,n\}\big\},$$

whose complement can be decided analogously to L_1 and is thus in coNP.

Hardness is shown via reduction from SAT-UNSAT, which is D^p-complete [PY82]. SAT-UNSAT asks whether for a pair $C_1 = \{C_1^1,\dots,C_1^{v_1}\}$ and $C_2 = \{C_2^1,\dots,C_2^{v_2}\}$ of Boolean formulae in CNF over variables $X^{C_i} = \{x_1^i,\dots,x_{n_i}^i\}$ for $i \in \{1,2\}$, C_1 is satisfiable and C_2 is unsatisfiable. We w.l.o.g. assume that $X^{C_1} \cap X^{C_2} = \emptyset$. Let

$$S^{C_i} \coloneqq (\{x_j, \neg x_j\})_{j\in\{1,\dots,n_i\}} + (C_i^k \cup \{p^{C_i}\})_{k\in\{1,\dots,v_i\}},$$

with $+$ denoting tuple concatenation.

As a supplementary statement, we show that the size of the MinCover π^{C_i} of S^{C_i} (denoted as $k_{min}^{S^{C_i}}$) to be equal to n_i if C_i is satisfiable and n_i+1 otherwise. For the proof, we first remark that $n_i \le k_{min}^{S^{C_i}} \le n_i + 1$. The lower bound follows from S^{C_i} containing n_i pairwise disjoint sets (of the form $\{x, \neg x \mid x \in X^{C_i}\}$), and the upper bound can be achieved by covering $t \in \{n_i+1,\dots,n_i+v_i\}$ via $\pi(t) = p^{C_i}$. Assume that C_i is satisfiable. Then, there exists by definition an $\alpha : X^{C_i} \to \{0,1\}$ where for all $C' \in C_i$ there is some $l \in C'$ s.t. $\alpha(l) = 1$ if $l \in X^{C_i}$ or $\alpha(x) = 0$ if $l \in \overline{X^{C_i}}$. We define

$$\pi^{C_i}(t) = \begin{cases} x_t & \text{if } 1 \le t \le n_i \text{ and } \alpha(x_t^i) = 1, \\ \neg x_t & \text{if } 1 \le t \le n_i \text{ and } \alpha(x_t^i) = 0, \text{ and} \\ l & \text{if } n_i + 1 \le t \le n_i + v_i \text{ for some } l \in C_i^t \text{ s.t.} \\ & \quad \alpha(l) = 1 \text{ if } l \in X^{C_i} \text{ or } \alpha(x) = 0 \text{ if } l \in \overline{X^{C_i}}. \end{cases}$$

This function satisfies $\pi^{C_i}(t) \in S_t$ for $t \in \{1,\dots,n_i+v_i\}$ and $|\mathrm{Im}(\pi^{C_i})| = n_i$ and is hence, due to the established lower bound, a MinCover. If C_i is unsatisfiable, $k_{min}^{S^{C_i}} \neq n_i$ as otherwise there must exist a MinCover π s.t. $\pi(t) \in \{x_t, \neg x_t\}$ for $t \in \{1,\dots,n_i\}$ and $\pi(n_i + t) \in C_i^t$ for $t \in \{1,\dots,v_i\}$. Note that $p^{C_i} \notin \mathrm{Im}(\pi)$, as π always selects n_i literals and hence it would hold that $|\mathrm{Im}(\pi)| \neq n_i$. We can now construct a satisfiable assignment for C^i by $\alpha(x_t) = 1$ iff $\pi(t) = x_t$ which contradicts unsatisfiability of C_i and hence $k_{min}^{S^{C_i}} \neq n_i$. From the established upper bound, it follows that $k_{min}^{S^{C_i}} = n_i + 1$.

We now define the polynomial-time reduction from CNFs C_1 and C_2 of SAT-UNSAT to MinCover as $S \coloneqq S^{C_1} \cup S^{C_1'} \cup S^{C_2}$, where C_1' a copy of C_1 with disjoint variables. Then, $k_{min}^{S} = 2n_1 + n_2 + 1$ iff C_1 is satisfiable and C_2 is unsatisfiable: Since all elements of S^{C_1}, $S^{C_1'}$, and S^{C_2} are disjoint, it holds that if

- C_1 unsatisfiable and C_2 unsatisfiable, the MinCover of S has size $2n_1 + n_2 + 3$,

- C_1 unsatisfiable and C_2 satisfiable, the MinCover of S has size $2n_1 + n_2 + 2$,
- C_1 satisfiable and C_2 unsatisfiable, the MinCover of S has size $2n_1 + n_2 + 1$,
- C_1 satisfiable and C_2 satisfiable, the MinCover of S has size $2n_1 + n_2$,

which shows the equivalence. □

While Theorem 5.4.1 shows that IsMinCover is computationally hard, there still exists a natural encoding into a problem for which optimized algorithms exist. Namely, we can rely on identifying minimal hitting sets for computing MinCover. A typical solution to minimal hitting set is via unweighted partial MaxSAT, where it is asked to find a solution to two CNFs (given in the notion of Boolean CNF introduced above):

1. C_h, the hard clauses that must be satisfied by the solution, and
2. C_s, the soft clauses, for which the number of satisfied clauses must be maximized by the solution.

We can now leverage a MaxSAT solver to generate a solution to MinCover by setting

$$C_h = S,\ C_s = \big\{\{\neg s\} \mid s \in U\big\}.$$

Intuitively, the soft clauses force the solver to maximally exclude elements of U (ensuring that the solution is minimal), while the hard clauses enforce to hit each set at least once. Then, $\kappa_{min} = |\{s \in U \mid \alpha(s) = 1\}|$, i.e., we return simply the number of satisfied variables.

In practice, MaxSAT solvers can offer satisfactory run time, despite the problem being NP-hard (specifically, OptP-complete [Kre88]). A worst-case complexity of $\mathcal{O}^*(1.2886^n)$ has been achieved by optimized MaxSAT algorithms [Xia22]. This offers a significant improvement over the brute-force approach to MinCover with a worst-case complexity of $\mathcal{O}(2^n)$.

Example 5.4.4

For our running example of Example 5.4.3, we generate the following CNFs from our four scenarios s_1 to s_4:

$$C_h = \big\{\{(t_4)\}, \{(t_1, t_4)\}, \{(t_4), (t_1, t_4)\}, \{(t_4), (t_2, t_4), (t_1, t_4), (t_1, t_2, t_4)\}\big\},$$
$$C_s = \big\{\{\neg(t_4)\}, \{\neg(t_1, t_4)\}, \{\neg(t_2, t_4)\}, \{\neg(t_1, t_2, t_4)\}\big\}.$$

We can see that the maximal solution must be $\alpha((t_4)) = \alpha((t_1, t_4)) = 1, \alpha((t_2, t_4)) = \alpha((t_1, t_2, t_4)) = 0$. Thus, we satisfy two of the four soft clauses, which confirms $\kappa_{min} = 2$.

We now show correctness of the sketched translation procedure.

Theorem 5.4.2 (Correctness of reducing MinCover to MaxSAT [Wes+26])

For a given MinCover instance $(S_1, \dots, S_n)$ over the universe $U = \bigcup_{i \in \{1,\dots,n\}} S_i$, $k_{min} = |\{s \in U \mid \alpha(s) = 1\}|$ where α is a solution of the MaxSAT instance $C_h = S$ and $C_s = \{\{\neg s \mid s \in U\}\}$.

Proof 5.4.2 (of Theorem 5.4.2 [Wes+26])

We are given a list of sets $(S_1, \dots, S_n)$ over the universe $U = \bigcup_{i \in \{1,\dots,n\}} S_i$. For the first direction, let π be a MinCover of S with $|\mathrm{Im}(\pi)| = k_{min}$. Then, define

$$\alpha_\pi(s) = \begin{cases} 1 & \text{if } s \in \mathrm{Im}(\pi), \\ 0 & \text{otherwise.} \end{cases}$$

α_π is a solution to the CNF $C_h = S$ as $\pi(i) \in S_i$ for all $i \in \{1, \dots, n\}$. Hence, for all optimal solutions α to the MaxSAT problem $C_h = S$ and $C_s = \{\{\neg s \mid s \in U\}\}$ it must hold that $|\{s \in U \mid \alpha(s) = 1\}| \leq |\{s \in U \mid \alpha_\pi(s) = 1\}| = |\mathrm{Im}(\pi)| = k_{min}$, as the soft clauses in C_s enforce minimization of the number of $s \in U$ s.t. $\alpha(s) = 1$.

Conversely, let α be a solution to the MaxSAT problem $C_h = S$ and $C_s = \{\{\neg s \mid s \in U\}\}$. Then, for all $i \in \{1, \dots, n\}$, define $\pi_\alpha(i) = s_i$ for some $s_i \in S_i$ with $\alpha(s_i) = 1$ (which exists as α satisfies the hard clauses S). Then, π_α is a cover of S since for each $i \in \{1, \dots, n\}$ it holds that $\pi_\alpha(i) \in S_i$. It follows that $k_{min} \leq |\mathrm{Im}(\pi_\alpha)| \leq |\{s \in U \mid \alpha(s) = 1\}|$.

Combining both inequalities $k_{min} \leq |\{s \in U \mid \alpha(s) = 1\}|$ and $|\{s \in U \mid \alpha(s) = 1\}| \leq k_{min}$ yields $k_{min} = |\{s \in U \mid \alpha(s) = 1\}|$. □

MaxCover

MaxCover is efficiently solvable, i.e., in polynomial time. This can be shown by reducing MaxCover to the problem of finding maximum matchings in bipartite graphs [HK71]. An undirected graph $G = (V, E)$ is bipartite if there exists a partition of V into V_1 and V_2 s.t. $E = \{\{v_1, v_2\} \mid v_1 \in V_1, v_2 \in V_2\}$. A maximum matching is then a maximal set of edges $E_{max} \subseteq E$ s.t. for all $\{v_1, v_2\}, \{v_3, v_4\} \in E_{max}$ with $\{v_1, v_2\} \neq \{v_3, v_4\}$ it holds that $v_1 \neq v_3$ and $v_2 \neq v_4$. Hence, we are searching for a maximal edge set without adjacent edges. This problem – which we call MaxMatch – is in P. For instance, the Hopcroft-Karp algorithm has a worst-case run time of $\mathcal{O}(\sqrt{V}E)$ [HK71]. Note that there are even more optimized approaches [CK24].

For the reduction of MaxCover to MaxMatch, the sets can be encoded as a graph

$$G_S = (\{1, \dots, n\} \cup U, \{\{i, s\} \mid i \in \{1, \dots, n\}, s \in S_i\}),$$

with partitions $\{1, \ldots, n\}$ and U. Then, the cardinality of a maximal matching in G_S is the desired κ_{max}, or, formally, $\kappa_{max} = |E_{max}|$. Intuitively, the matching maximally selects elements from each set (based on the membership constraints encoded as edges) and ensures that at least one element is chosen for each set while "blocking" already selected elements as choices for other sets (by disallowing adjacent edges). In terms of the inputs to MaxCover, the Hopcroft-Karp algorithm achieves a worst-case complexity of $\mathcal{O}(\sqrt{n + |U|} \sum_{S_i \in S} |S_i|)$. This implies IsMaxCover to be in P. We hence improve over both the brute-force and SMT-based procedure in theoretical complexity. It remains to show correctness of the reduction.

Theorem 5.4.3 (Correctness of reducing MaxCover to MaxMatch [Wes+26])

For a given MaxCover instance $(S_1, \ldots, S_n)$ over the universe $U = \bigcup_{i \in \{1,\ldots,n\}} S_i$, $k_{max} = |E_{max}|$ where E_{max} is a maximum matching in the bipartite graph $G_S = (\{1, \ldots, n\} \cup U, \{\{i, s\} \mid i \in \{1, \ldots, n\}, s \in S_i\})$.

Proof 5.4.3 (of Theorem 5.4.3 [Wes+26])

We are given a list $(S_1, \ldots, S_n)$ of sets over the universe $U = \bigcup_{i \in \{1,\ldots,n\}} S_i$. For the first direction, let π be a MaxCover of S with $|\mathrm{Im}(\pi)| = k_{max}$. Then, $E_\pi = \{\{i_s, s\} \mid s \in \mathrm{Im}(\pi)\}$ with i_s an arbitrary element from $\pi^-(s)$ is a matching in the bipartite graph $G_S = (\{1, \ldots, n\} \cup U, E)$ with $E = \{\{i, s\} \mid i \in \{1, \ldots, n\}, s \in S_i\}$: First, $E_\pi \subseteq E$. Second, for two different edges $\{i_{s_1}, s_1\} \in E_\pi$ and $\{i_{s_2}, s_2\} \in E_\pi$ it holds that $i_{s_1} \neq i_{s_2}$ (as π is a function) and $s_1 \neq s_2$ (as we selected only one element from $\pi^-(s)$ for each $s \in \mathrm{Im}(\pi)$). Thus, E_π is a matching of size k_{max} and hence for any maximum matching E_{max} it must hold that $k_{max} = |E_\pi| \leq |E_{max}|$.

Conversely, let E_{max} be a maximum matching in the bipartite graph $G_S = (\{1, \ldots, n\} \cup U, \{\{i, s\} \mid i \in \{1, \ldots, n\}, s \in S_i\})$. Then, $\pi_{E_{max}} : \{1, \ldots, n\} \to U$ with

$$\pi_{E_{max}}(i) = \begin{cases} s \text{ such that } \{i, s\} \in E_{max} & \text{if such } s \text{ exists,} \\ \text{any } s \in S_i & \text{otherwise,} \end{cases}$$

is a cover of S since for each $i \in \{1, \ldots, n\}$ it holds that $\pi(i) \in S_i$. Note that $|\mathrm{Im}(\pi_{E_{max}})| \geq |E_{max}|$ and hence for any MaxCover π it must hold that $|E_{max}| \leq |\mathrm{Im}(\pi_{E_{max}})| \leq |\mathrm{Im}(\pi)| = k_{max}$.

Combining both inequalities $k_{max} \leq |E_{max}|$ and $|E_{max}| \leq k_{max}$ yields $k_{max} = |E_{max}|$.□

Example 5.4.5

Figure 5.9 shows the bipartite graph for the four scenarios given in Example 5.4.3. The maximal matching is represented as dashed lines.

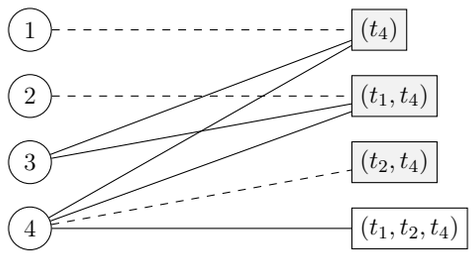

Figure 5.9: The bipartite graph G_S for the four scenarios of Example 5.4.3. The left partition represents the sets S_1 to S_4, and the right partition contains their elements. The maximum matching is given as dashed lines.

Note that the third set does not require a custom entry in the matching, as all its elements are already covered by some other match (namely, by the first and second set).

5.4.4 Evaluation

Synthetic Data

The main purpose of integrating MTCQs into coverage computation is to answer whether MTCQs and their implied OWA can offer benefits in a practical context. We have seen above that this OWA has profound theoretical implications. Two urgent questions arise:

SQ1 Can both bounds even be computed under realistic data sizes and tag configurations, and

SQ2 do they offer meaningful improvements on naïve bound approximations?

If this were not the case, MTCQ-based coverage would offer little advantages to safety engineers. The claim of this dissertation would then not be supported by this section.

The nature of these two questions demands controlled, synthetic data, allowing us to assess the impact of different variables onto the outcomes of interest. We rely on the data publicly provided by Schmid and Westhofen [SW25], which were created by generating data for $n_t \in [1, 15]$ tags and Boolean values distributed fairly. Initially, these traces do not contain $*$-entries. For an amount $n_{owa} \leq n_t$ of OWA tags, the original values were overwritten by $*$ with a random chance $p_* \in \{0.05, 0.1, 0.15, 0.2\}$. The remaining CWA tags ($n_{cwa} = n_t - n_{owa}$ many) were unchanged. This enables reporting the "real coverage" in the following. For larger n_t, the data are generated batch-wise for efficiency reasons, and the batch size grows correspondingly to the number of assessed total tags (the largest batch for $n_t \geq 13$ contains 10 000 scenarios). An example of the generated data is shown in Figure 5.10.

All computations on the synthetic data were performed on an Intel® Xeon® Gold 6342 CPU (single-threaded). The experiment data contain only a single run for each

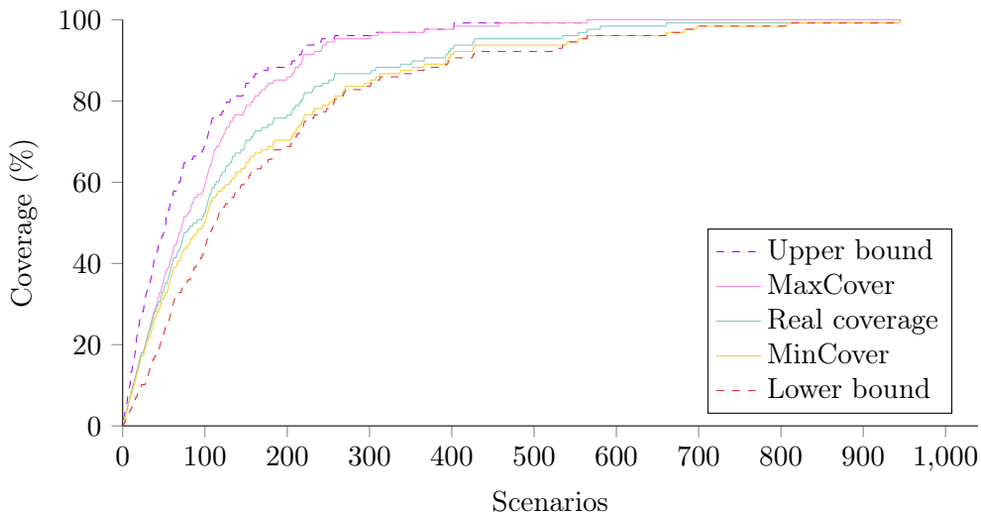

Figure 5.10: Real coverage, MinCover, MaxCover, and naïve bounds for $n_{\text{owa}} = 3$, $n_{\text{cwa}} = 4$, and $p_* = 0.1$.

configuration, as experience showed results to not differ meaningfully. MinCover was implemented using Z3 [MB08] and the above described reduction to MaxSAT. Our MaxCover implementation relies on Edmonds' blossom or the Hopcroft-Karp algorithms for maximum cardinality (bipartite) matching [Mic+20], one of which has to be selected by the user.

For question SQ1 – concerning run time – we assess the dependency of the size of the dataset on the wall clock run time of both MinCover and MaxCover in Figure 5.11 at the largest configuration for which all values of p_* delivered results ($n_{\text{owa}} = 13$, $n_{\text{cwa}} = 0$) without having to be aborted due to excessive run time. Almost all of the time is spent solving MinCover: MaxCover is only computed up to, at most, $5 \cdot 10^4$ scenarios, with a negligible computing time of, on average, 248.5 ms at its last iteration (for Edmond's blossom). In general, a sublinear dependency is evident from the results. This answer positively question SQ1 – even on large datasets, coverage approximations can be computed efficiently, and we successfully avoided an exponential dependency on the data (in contrast to the above presented brute-force approach). Spikes at the beginning of Figure 5.11 are likely originating from solver initialization.

An additional concern is the dependency of run time on the number of tags, for which we hypothesized an exponential relation. Figure 5.12 shows the wall clock run time of MinCover over different numbers of OWA tags over the four probability configurations in the final iteration. Note that MaxCover reached 100% well before that point and its run time is therefore zero at that iteration. The exponential dependency is obvious from Figure 5.12. The most likely explanation relates to the number of scenarios that have to

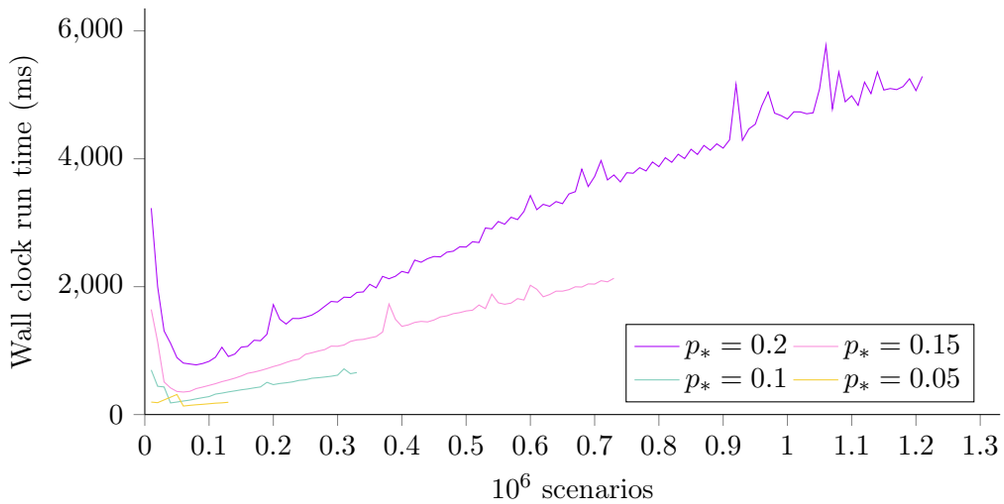

Figure 5.11: Wall clock run time of MinCover and MaxCover for $n_{\text{owa}} = 13$ and $n_{cwa} = 0$.

be considered at 100% coverage: At worst, if all tags are interpreted under an OWA, there can be $(|V| + 1)^{|T|}$ many different hard clauses (in the experiments, this translates to 3^{n_t}). Therefore, we can hypothesize that the exponential dependency of run time on the number of tags is caused by adding a new hard clause to the MaxSAT problem for each new scenario. For example, at $n_{owa} = 15$ the solver can, at worst, consider 3^{15} different hard clauses. In our experiments, the computation terminated at an order of magnitude of 10^6 hard clauses.

The second question, SQ2, is concerned with the approximation quality of MinCover and MaxCover. For this, we define naïve lower and upper bounds on coverage: The upper bound (b_u) assumes that all options of the scenario happened, which is obviously impossible. For the lower bound (b_l), we throw out all scenarios with $*$-entries. We report how much MinCover and MaxCover reduce the error from the naïve bounds to real coverage r in Figure 5.13. For MinCover, this error reduction is defined as

$$\frac{\kappa_{min} - b_l}{r - b_l}.$$

The error for MaxCover is given analogously.

We can see that, on average, meaningful error reductions are made by MinCover and MaxCover, which is stronger in the first half: The error is reduced by 55.92% (MinCover) and 59.68% (MaxCover). In the second half, this regresses to an average error reduction of 27.89% (MinCover) and 16.46% (MaxCover). Overall, these results indicate that MinCover and MaxCover can meaningfully improve coverage estimations. While MaxCover has its best performance at lower data sizes (thus, at earlier development stages), MinCover also shows nontrivial error improvements at later phases. Fortunately, this is in line with

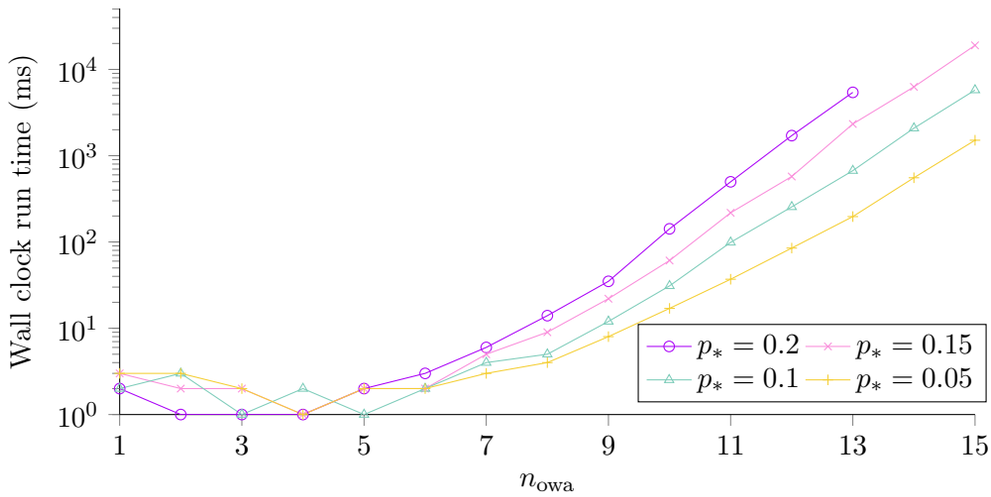

Figure 5.12: Log-scaled wall clock run time of the last iteration for computing MinCover at different amounts of OWA tags. For $p_* = 0.2$ and $n_{\text{owa}} > 13$, experiments had to be aborted early due to excessive run time.

our interpretation of their practical utilization: MaxCover can be seen most useful in supporting engineers during short-lived, early development cycles (where real coverage is typically low): It allows quantifying the definitely still required testing efforts, on which engineers can base decisions regarding future test campaigns, e.g., to increase simulative testing or change the circumstances of public test drives. MinCover, on the other hand, gives worst-case guarantees and can therefore be used as an argument towards approval authorities, as required in Germany [Ver22b]. Hence, the role of MinCover becomes more significant as the test campaign progresses. Our results show that MinCover is still able to deliver a meaningful approximation also at that stage.

Real-World Data

The previous analysis used solely synthetic data which were created independent of MTCQs. We used it to positively answer questions on quality and performance of the presented OWA coverage algorithms. However, it did not rely on MTCQs but rather assumed $*$-entries being present in the data. As a final step, we next evaluate the extent to which MTCQs support the assumed OWA coverage setting:

RQ1 Can Topllet (which answers MTCQs) be integrated into a state-of-the-art coverage software?

RQ2 Can MTCQs be efficiently computed on real-world data to induce gray zones in tag values?

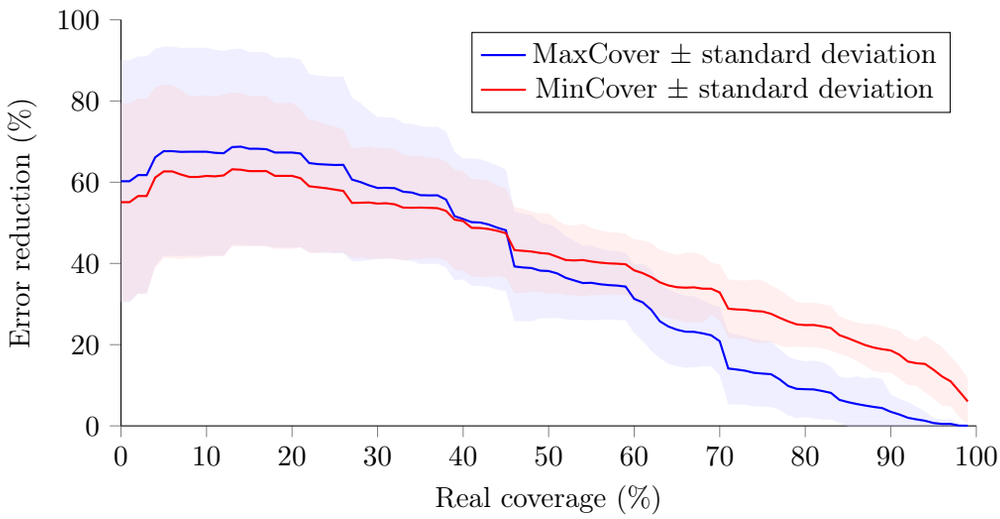

Figure 5.13: Mean error reduction of all experiments with $n_{\text{owa}} \geq 5$ and $p_* = 0.05$ over the achieved real coverage, with the standard deviation depicted as a colored area.

RQ3 Do these gray zones imply similar $*$-entry distributions assumed in the above synthetic experiments?

For RQ1, we provide a publicly available integration of Topllet into STARS, a software tool to classify scenario data and compute coverage on the resulting values. STARS offers functionality to represent recorded scenarios in user-definable data models, evaluate (temporal) formulae on it, and compute several metrics, including coverage, over the results. Its default CWA logic for scenario classification is CMFTBL, a metric first-order temporal logic. In its core, however, STARS can be seen independent of CMFTBL, as it allows for arbitrary computable functions to determine values for each tag given the data. To now incorporate unknown tag values into STARS, exclusion and inclusion criteria were added for each tag, as described by Westhofen et al. [Wes+26]. The custom module `stars-logic-mtcq` then integrates MTCQs by the mechanism present in the following [Wes25b].

We now show how MTCQs are formally integrated into STARS. Recall that we are tasked with deriving from data d a scenario possibly containing unknown information, i.e., implementing the function $\Lambda_{T,V\cup\{*\}} : \bigcup_{h>0} \mathbb{D}^h \to (V \cup \{*\})^T$ that was introduced above. In order to enable DL reasoning, each data trace d is given as a TKB $d = (\mathcal{O}, (\mathcal{D}_j)_{j\in\{0,\dots,m\}})$. This means that an additional ontology $\mathcal{O}$ must be provided by the user. We can now formulate $\Lambda_{T,V\cup\{*\}}$ in terms of positive and negative (possibly non-Boolean) MTCQs Φ_t^- and Φ_t^+ representing answers that safely satisfy and violate a tag $t \in T$, respectively. While answers to the negative formula provide a verdict on the safe violation of the tag,

answers to the positive formula represent a guaranteed tag satisfaction. An additional third query Φ_t° allows restricting the search space to a subset of individuals in d. This allows the user to specify relevant individuals for the tag (such as vehicles for a tag concerned with passing of parking vehicles). Otherwise, the gray zone might be too large and, in practice, $*$ can be assigned too liberally. Formally, we define $\Lambda_{T,V\cup\{*\}}$ as

$$\Lambda_{T,V\cup\{*\}}(d) = t \mapsto \begin{cases} 0, & \text{if } \mathsf{cert}_d(\Phi_t^-) = \mathsf{cert}_d(\Phi_t^\circ), \\ 1, & \text{if } \mathsf{cert}_d(\Phi_t^+) \cap \mathsf{cert}_d(\Phi_t^\circ) \neq \emptyset, \\ *, & \text{otherwise.} \end{cases} \tag{5.2}$$

Intuitively, if *all* individuals in the search space are guaranteed to satisfy the exclusion criterion, 0 can be assigned with certainty. If there is *some* individual for which a tag satisfaction (w.r.t. the search space) can be inferred, the tag value is 1. Otherwise, there must be some individual in the search space for which neither the exclusion nor the inclusion criterion is satisfied, and we assign $*$. In the special case of Boolean queries, Φ_t° can be set to true achieve the desired semantics (since $\mathsf{cert}_d(\mathsf{true}) = \{\emptyset\}$). As a simplifying assumption in the above formalism, we set $V = \{0, 1\}$ and can therefore count multiple occurrences of a tag in a given data record only once.

The above formalism was implemented as a custom module `stars-logic-mtcq` in the STARS framework [Wes25b]. Its general workflow is shown in Figure 5.14. The module allows converting arbitrary data models in STARS via OWLAPI into the TKB data model of Topllet, either via annotation (`@DLConvertible`) of the to-be-converted classes or by explicit enumeration. Additionally, the user must specify an ontology file that is read via OWLAPI. The ontology axioms are then considered during reasoning. After all three queries (search space, inclusion, and exclusion criteria) are answered, the tag is evaluated according to Equation (5.2). An example is provided in the referenced artifact. Our module provides evidence for the fact that Topllet can be used in combination with a state-of-the-art coverage framework (RQ1).

For RQ2, we require evaluating the efficacy of using MTCQs on real-world data resembling a realistic ODD. We rely on the four example tags from Example 5.4.1 representing potential triggering conditions and hazardous behaviors. For the ontology, we employ a reduced and slightly adapted version of the already introduced A.U.T.O. It contains as formal axioms what has been intuitively described in Example 5.4.2. We use the inD dataset to compute the tag values. The data encompass 7 538 scenarios at four intersections in Aachen, Germany, taken by a drone camera [Boc+20]. Trajectories and traffic participant classifications are extracted by means of neural networks and enriched with high definition road network information. The data stream is split into single scenarios by considering each passenger car as the unique ego vehicle during its presence within the camera frame. We sampled it with two Hertz and converted it to OWL2-files corresponding to the TKB input format of Topllet.

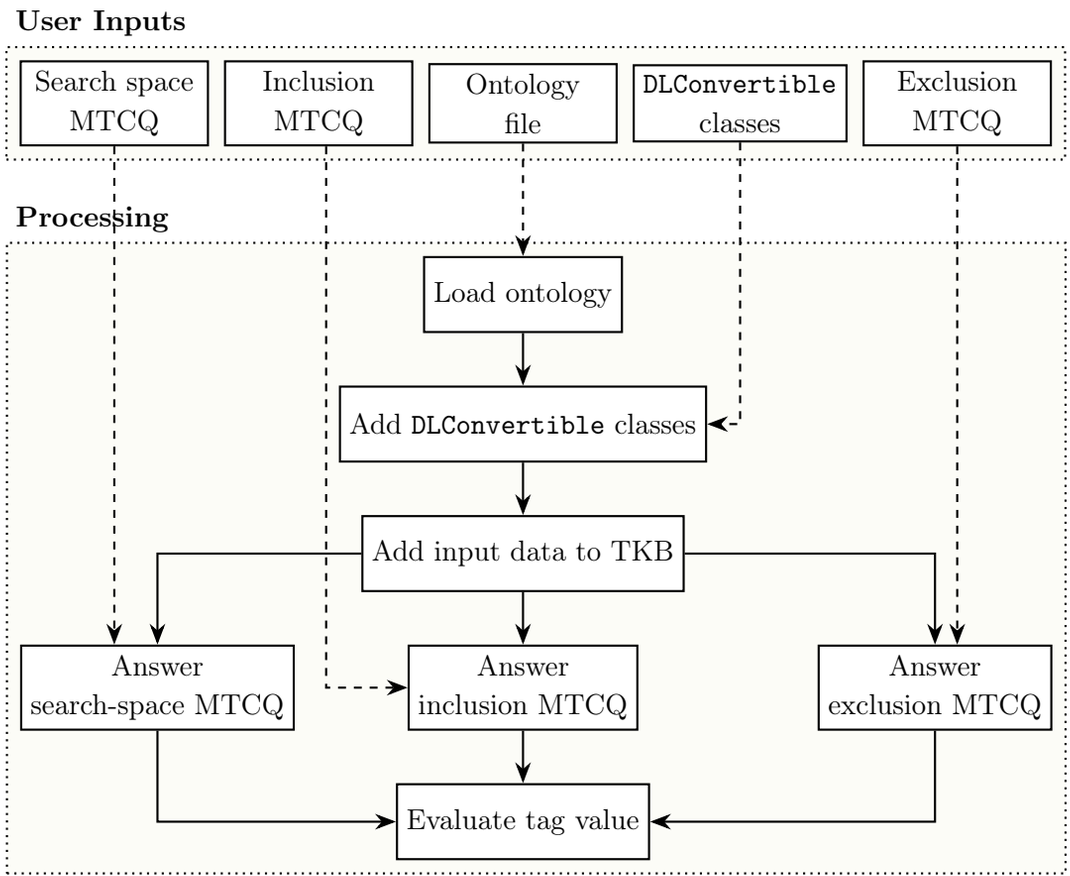

Figure 5.14: Workflow of the `stars-logic-mtcq` module.

We have executed the analysis of inD on a virtual machine with access to 20 cores of an AMD EPYC™ 7542 with 128 GB of memory. The evaluation artifact is available online [Wes26c], and also contains the employed queries and ontology. Unfortunately, for reproduction, an individual copy of the data must be obtained due to licensing. We have measured total wall clock run time of Topllet configured with the rewriting-based algorithm (which includes JVM and tool initialization, parsing, loading of TKBs, query rewriting and execution, as well as result output).

On average, for the 7 538 scenarios of inD, answering each query has taken 6.11 seconds with the median being 4.44 seconds. Note that each tag implies executing three MTCQs, i.e., in the experiments, we have run Topllet 90 456 times. While a minority of selected scenarios have induced a comparably high run time on some queries (showing that edge case optimization is still possible), overall, 95% of the queries could be answered within 11.22 seconds. Similarly, the 90th and 99th percentiles are 8.41 and 23.54 seconds, respectively. This highlights that MTCQs can be answered efficiently on inD, and positively answers

Table 5.6: Average answering times for the search space, negative, and positive queries for each tag on inD as well as the average for each tag and query (∅), in seconds.

Ψ \ t	t_1	t_2	t_3	t_4	∅
Ψ_t°	4.66s	4.65s	4.64s	4.65s	4.65s
Ψ_t^-	5.03s	9.08s	5.36s	5.12s	6.15s
Ψ_t^+	5.62s	6.11s	5.24s	13.12s	7.52s
∅	5.10s	6.61s	5.08s	7.63s	6.11s

Q2 – Topllet can be used on real-world data.

The final question, RQ3, regards the implied tag value distribution, which we present in Table 5.7. The results indicate that $*$ entries of the four example tags occur with a

Table 5.7: Distributions for the OWA example tags on inD.

V_t \ t	t_1	t_2	t_3	t_4
0	69.49%	46.46%	61.95%	72.19%
1	29.99%	39.59%	29.01%	0.34%
$*$	0.52%	13.96%	9.03%	27.46%

probability between 0.52% to 27.46%. This provides evidence for the validity of above-made assumptions, where, for the synthetic data, values for p_* in $\{0.05, 0.1, 0.15, 0.2\}$ were used. While higher probabilities than $p_* = 0.2$ are certainly possible (cf. t_4), the selection appears to match the general trend extractable from Table 5.7.

In summary, we have examined integrability of MTCQs into a broader domain context, namely, scenario-based coverage. We have highlighted the profound theoretical and practical implications of transitioning from the typical CWA to the underlying OWA of MTCQs. This has opened up challenging new perspectives on scenario coverage, for which we hope to spark interest in the research community. Our practical demonstration has included an integration of Topllet into STARS as well as an application of Topllet on real-world drone data. While the current state of Topllet can be seen as a promising step towards industrial application of temporal querying, the next section delineates several lines of work which can yield further improvements in performance and usability.

6 Conclusion

Summary

In overall terms, the objective of this dissertation was to show that

> "temporal querying in expressive DLs can be made practically applicable, specifically for assessing safety-critical scenarios in road traffic."

Through various chapters, we have systematically analyzed the application context, devised a suitable query language, developed algorithms to answer these queries, and demonstrated practical applicability by means of custom benchmarks and implementations.

We started with a thorough examination of automated driving safety, specifically, SOTIF. An improvement of the corresponding standard, ISO 21448, and a literature review on scenario-based testing enabled us to derive a practically relevant ontology-based temporal query language. This language is based on answering CQs over expressive DLs. We delineated how state-of-the-art tools can offer a highly optimized performance despite ExpTime-completeness of the associated decision problems.

We devised two optimized algorithms to efficiently answer such queries, showed correctness, and analyzed their computational complexity. While the first algorithm relies on FAs, the second operates directly on the formulae via rewriting into a normal form. A theoretical analysis revealed that, on a nontrivial fragment of queries, no UCQs are required for answering. When using deterministic FAs, this is not always the case. This behavior, however, is avoided by the rewriting-based algorithm.

Both algorithms were implemented in a software tool, and we provided a custom benchmark generator for our application setting of automated driving. The evaluation confirmed the above theoretical hypothesis by showing roughly an order of magnitude performance improvement by the rewriting-based algorithm, mostly originating from fewer UCQ checks. Besides performance, we also considered the implications of using a DL-based temporal logic in a practical setting. We have seen that replacing standard

temporal logics with MTCQs in scenario-based test coverage, an important concept in SOTIF, enables assumptions more akin to those required in real-world environments. The theoretical implications on coverage were profound; fortunately, useful approximations were still possible.

Future Research Directions

Future research roughly follows the three possible perspectives on the topic of this dissertation: Practice, performance, and theory.

Improving usability We devised our language with practical applicability in mind, and it is therefore only natural to further explore topics related to usability. Specifically, the following endeavors can be fruitful:

- Whereas this dissertation focused on adding temporal capabilities to DL querying, its practical importance could be further enhanced by widening the scope to spatial reasoning. This includes, for example, features akin to those of the Region Connection Calculus. While work exists on adding spatial capabilities to DLs, we are not aware of an expressive DL querying mechanism that integrates both temporal and spatial inference.
- In practice, the recursive structure of LTL-based logics can be confusing to engineers. A possible solution is the use of a domain-specific language that can be internally translated into MTCQs. Ideally, this would be combined with a possibility for visual specification in a graphical user interface, and extensive empirical user testing. Similarly, additional features (like the prevalence operators of CMFTBL) can further enhance usefulness of the query language.
- In industrial settings, temporal data are often stored in time series databases, such as InfluxDB or TimescaleDB. An integration of MTCQs into such tooling could offer an easy entrance to DL-based temporal querying for practitioners.
- Adding past temporal operators to MTCQs widens their use towards online monitoring applications. This allows running them on the automated system itself, e.g., for situation understanding or diagnostic purposes. While a trivial solution is to convert such formulae to pure future-time ones, it is not directly apparent how the rewriting approach can be adapted to formulae with nested past- and future-time operators.
- A minor point concerns the employed OWL2-based TKB file format. Such files can grow excessively large as each timestamp contains all required information. A more efficient approach would be to only store a delta to the previous timestamp, or

use a single file containing data axioms annotated with intervals. Both approaches will, however, break with existing tooling around OWL2, requiring additional implementation effort.

Optimizing performance While we observed adequate performance on our benchmarks, more involved settings (such as the mentioned online monitoring) will most likely require even further improvements.

- The most obvious area for performance improvements concerns incremental temporal reasoning. However, first experiments with the incremental reasoning capabilities currently implemented in Openllet showed discouraging results, where computing the axiom difference between two input ontologies took away more time than what was gained through avoiding additional reasoning steps. This line of future work is closely linked with transitioning towards a delta-based file format, which could alleviate the observed issue.

- CQ answering has been made highly efficient. We have seen at several points that UCQs act as the performance bottleneck, and, even in the rewriting-based algorithms, they cannot always be avoided. It is therefore imperative to further improve UCQ answering from a practical perspective, similar what has been done for CQs. In general, established techniques like query reordering should be transferrable.

Advancing theory Lastly, we left some theoretical questions open that were out of scope for this dissertation. Their pursuit might offer deeper insights in the behavior of temporal queries over expressive DLs.

- Can we be even more sparingly with UCQ checks than the rewriting-based algorithm? It can easily be seen that some UCQs that are answered by the algorithm can be collapsed into a CQ (such as $\mathtt{A}(\mathtt{a}) \vee \mathtt{A}(\mathtt{a})$), however, a general procedure for doing so appears to be unknown. If we employed such a hypothetical UCQ-collapsing procedure, could we prove the rewriting-based approach to be optimal w.r.t. the UCQ checks it performs?

- While the definition of $\mathrm{TCQ}_{\mathrm{CQ}^*}$ has fully served its purpose in the context of this dissertation (uncovering unnecessary UCQ checks), some theoretical details remain subject to further investigation. First, more classes of queries can be added to $\mathrm{TCQ}_{\mathrm{CQ}^*}$, e.g., by allowing a safe form of negation. How far can we broaden the syntactic fragment? The question arises whether it is possible to cover, in a semantical sense, $\mathrm{TCQ}_{\mathrm{CQ}}$ with such a syntactic definition (meaning that there is for each query in $\mathrm{TCQ}_{\mathrm{CQ}}$ a semantically equivalent query in the syntactic fragment)? On a related note, we have shown that deciding membership in $\mathrm{TCQ}_{\mathrm{CQ}^*}$ is efficiently decidable. Does the same hold for $\mathrm{TCQ}_{\mathrm{CQ}}$?

These possible future endeavors highlight that we have just begun exploring temporal querying in expressive DLs, and much can be gained from further promoting the topic of this dissertation.

Bibliography

[Ada94] L. D. Adams. *Review of the literature on obstacle avoidance maneuvers: braking versus steering*. Technical Report. Ann Arbor, MI, USA: University of Michigan, Transportation Research Institute, Aug. 1994. URL: https://hdl.handle.net/2027.42/1068 (cit. on pp. 46, 47).

[AF00] Alessandro Artale and Enrico Franconi. "A survey of temporal extensions of description logics". In: *Annals of Mathematics and Artificial Intelligence* 30.1 (June 2000), pp. 171–210. ISSN: 1573-7470. DOI: 10.1023/A:1016636131405 (cit. on p. 86).

[AFI14] Alexandre Armand, David Filliat, and Javier Ibañez-Guzmán. "Ontology-based context awareness for driving assistance systems". In: *IEEE Intelligent Vehicles Symposium, IV 2014, Dearborn, MI, USA, June 8-11, 2014*. New York, NY, USA: IEEE, 2014, pp. 227–233. DOI: 10.1109/IVS.2014.6856509 (cit. on pp. 48, 51).

[AHR15] Rob Alexander, Heather Rebecca Hawkins, and Andrew John Rae. *Situation coverage–a coverage criterion for testing autonomous robots*. Technical Report. York, UK: Department of Computer Science, University of York, 2015. URL: https://eprints.whiterose.ac.uk/id/eprint/88736/ (cit. on p. 163).

[Aka+19] Yasuhiro Akagi, Ryosuke Kato, Sou Kitajima, Jacobo Antona-Makoshi, and Nobuyuki Uchida. "A Risk-index based Sampling Method to Generate Scenarios for the Evaluation of Automated Driving Vehicle Safety". In: *22nd IEEE Intelligent Transportation Systems Conference, ITSC 2019, Auckland, New Zealand, October 27-30, 2019*. New York, NY, USA: IEEE, 2019, pp. 667–672. DOI: 10.1109/ITSC.2019.8917311 (cit. on p. 37).

[All83] James F. Allen. "Maintaining Knowledge about Temporal Intervals". In: *Communications of the ACM* 26.11 (1983), pp. 832–843. DOI: 10.1145/182.358434 (cit. on p. 86).

[AMO18] Alessandro Artale, Andrea Mazzullo, and Ana Ozaki. "Temporal Description Logics over Finite Traces". In: *Proceedings of the 31st International Workshop on Description Logics, DL 2018, Tempe, Arizona, US, October 27-29, 2018*. Ed. by Magdalena Ortiz and Thomas Schneider. Vol. 2211. CEUR Workshop

Proceedings. Aachen, Germany: CEUR-WS.org, 2018. URL: https://ceur-ws.org/Vol-2211/paper-06.pdf (cit. on pp. 3, 86).

[Ani+12] Darko Anicic, Sebastian Rudolph, Paul Fodor, and Nenad Stojanovic. "Stream reasoning and complex event processing in ETALIS". In: *Semantic Web* 3.4 (2012), pp. 397–407. DOI: 10.3233/SW-2011-0053 (cit. on p. 86).

[Arc05] Jeffery Archer. "Indicators for traffic safety assessment and prediction and their application in micro-simulation modelling: A study of urban and suburban intersections". PhD thesis. Stockholm, Sweden: KTH Royal Institute of Technology, 2005. ISBN: 91-7323-119-3. URL: https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-143 (cit. on p. 40).

[Aré19] Nikos Aréchiga. "Specifying Safety of Autonomous Vehicles in Signal Temporal Logic". In: *2019 IEEE Intelligent Vehicles Symposium, IV 2019, Paris, France, June 9-12, 2019*. New York, NY, USA: IEEE, 2019, pp. 58–63. DOI: 10.1109/IVS.2019.8813875 (cit. on pp. 48, 50).

[Art+02] Alessandro Artale, Enrico Franconi, Frank Wolter, and Michael Zakharyaschev. "A Temporal Description Logic for Reasoning over Conceptual Schemas and Queries". In: *Logics in Artificial Intelligence, European Conference, JELIA 2002, Cosenza, Italy, September 23-26, 2002*. Ed. by Sergio Flesca, Sergio Greco, Nicola Leone, and Giovambattista Ianni. Vol. 2424. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2002, pp. 98–110. DOI: 10.1007/3-540-45757-7_9 (cit. on p. 84).

[Art+09] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. "The DL-Lite Family and Relations". In: *Journal of Artificial Intelligence Research* 36 (2009), pp. 1–69. DOI: 10.1613/JAIR.2820 (cit. on pp. 85, 121).

[Art+13] Alessandro Artale, Roman Kontchakov, Frank Wolter, and Michael Zakharyaschev. "Temporal Description Logic for Ontology-Based Data Access". In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013, Beijing, China, August 3-9, 2013*. Ed. by Francesca Rossi. Palo Alto, USA: IJCAI/AAAI Press, 2013, pp. 711–717. URL: https://www.ijcai.org/Proceedings/13/Papers/112.pdf (cit. on p. 85).

[Art+14] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. "A Cookbook for Temporal Conceptual Data Modelling with Description Logics". In: *ACM Transactions on Computational Logic* 15.3 (2014), 25:1–25:50. DOI: 10.1145/2629565 (cit. on p. 86).

[Art+17] Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. "Ontology-Mediated Query Answering over Temporal Data: A Survey". In: *24th International Symposium on Temporal Representation and Reasoning, TIME 2017, Mons, Belgium, October 16-18, 2017*. Ed. by Sven Schewe, Thomas Schneider, and Jef Wijsen. Vol. 90. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 1:1–1:37. DOI: 10.4230/LIPIcs.TIME.2017.1 (cit. on p. 87).

[Art+22] Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. "First-Order Rewritability and Complexity of Two-Dimensional Temporal Ontology-Mediated Queries". In: *Journal of Artificial Intelligence Research* 75 (2022), pp. 1223–1291. DOI: 10.1613/jair.1.13511 (cit. on p. 85).

[ASA21] ASAM e.V. *OpenSCENARIO XML V1.1.0*. Standard. Hoehenkirchen, Germany, Mar. 2021. URL: https://www.asam.net/standards/detail/openscenario-xml/older/ (cit. on p. 36).

[ASA22] ASAM e.V. *OpenXOntology User Guide V1.0.0*. Standard. Hoehenkirchen, Germany, Jan. 2022. URL: https://www.asam.net/standards/asam-openxontology/ (cit. on p. 53).

[ASA24a] ASAM e.V. *Open Simulation Interface V3.7.0*. Standard. Hoehenkirchen, Germany, July 2024. URL: https://www.asam.net/standards/detail/osi/ (cit. on p. 38).

[ASA24b] ASAM e.V. *OpenDRIVE V1.8.1*. Standard. Hoehenkirchen, Germany, Nov. 2024. URL: https://www.asam.net/standards/detail/opendrive/ (cit. on p. 38).

[ASA24c] ASAM e.V. *OpenSCENARIO DSL V2.1.0*. Standard. Hoehenkirchen, Germany, Mar. 2024. URL: https://www.asam.net/standards/asam-openxontology/ (cit. on pp. 35, 36).

[ASA24d] ASAM e.V. *OpenSCENARIO XML V1.3.1*. Standard. Hoehenkirchen, Germany, Nov. 2024. URL: https://www.asam.net/standards/detail/openscenario-xml/ (cit. on pp. 35, 36).

[ASC78] Brian L. Allen, B. Tom Shin, and Peter J. Cooper. "Analysis of Traffic Conflicts and Collisions". In: *Transportation Research Record* 667 (1978), pp. 67–74. URL: https://onlinepubs.trb.org/Onlinepubs/trr/1978/667/667-009.pdf (cit. on pp. 40, 45).

[AW19a] Christian Amersbach and Hermann Winner. "Defining Required and Feasible Test Coverage for Scenario-Based Validation of Highly Automated Vehicles". In: *22nd IEEE Intelligent Transportation Systems Conference, ITSC 2019, Auckland, New Zealand, October 27-30, 2019*. New York, NY, USA: IEEE, 2019, pp. 425–430. DOI: 10.1109/ITSC.2019.8917534 (cit. on p. 163).

[AW19b] Christian Amersbach and Hermann Winner. "Functional decomposition—A contribution to overcome the parameter space explosion during validation of highly automated driving". In: *Traffic Injury Prevention* 20.S1 (2019), S52–S57. DOI: 10.1080/15389588.2019.1624732 (cit. on p. 37).

[Baa+07] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, eds. *The Description Logic Handbook: Theory, Implementation, and Applications*. 2nd ed. Cambridge, UK: Cambridge University Press, 2007. ISBN: 0-521-87625-7. DOI: 10.1017/CBO9780511711787 (cit. on pp. 2, 61, 65–68, 75).

[Baa+20] Franz Baader, Stefan Borgwardt, Patrick Koopmann, Ana Ozaki, and Veronika Thost. "Metric Temporal Description Logics with Interval-Rigid Names". In: *ACM Transactions on Computational Logic* 21.4 (2020), 30:1–30:46. DOI: 10.1145/3399443 (cit. on p. 87).

[Bab+23] Stefan Babisch, Christian Neurohr, Lukas Westhofen, Stefan Schoenawa, Henrik Liers, et al. "Leveraging the GIDAS Database for the Criticality Analysis of Automated Driving Systems". In: *Journal of Advanced Transportation* 2023.1 (2023), p. 1349269. DOI: 10.1155/2023/1349269 (cit. on pp. 29, 32).

[Ban+20] Suguman Bansal, Yong Li, Lucas M. Tabajara, and Moshe Y. Vardi. "Hybrid Compositional Reasoning for Reactive Synthesis from Finite-Horizon Specifications". In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. Palo Alto, CA, USA: AAAI Press, 2020, pp. 9766–9774. DOI: 10.1609/AAAI.V34I06.6528 (cit. on p. 103).

[Bar+09] Evdoxios Baratis, Euripides G. M. Petrakis, Sotiris Batsakis, Nikolaos Maris, and Nikos Papadakis. "TOQL: Temporal Ontology Querying Language". In: *Advances in Spatial and Temporal Databases, 11th International Symposium, SSTD 2009, Aalborg, Denmark, July 8-10, 2009*. Ed. by Nikos Mamoulis, Thomas Seidl, Torben Bach Pedersen, Kristian Torp, and Ira Assent. Vol. 5644. Lecture Notes in Computer Science. Berlin, Germany:

Springer, 2009, pp. 338–354. DOI: 10.1007/978-3-642-02982-0_22 (cit. on p. 84).

[BBL05] Franz Baader, Sebastian Brandt, and Carsten Lutz. "Pushing the EL Envelope". In: *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, IJCAI 2005, Edinburgh, Scotland, UK, July 30 - August 5, 2005*. Ed. by Leslie Pack Kaelbling and Alessandro Saffiotti. Denver, CO, USA: Professional Book Center, 2005, pp. 364–369. URL: http://ijcai.org/Proceedings/05/Papers/0372.pdf (cit. on p. 85).

[BBL13] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. "Temporalizing Ontology-Based Data Access". In: *24th International Conference on Automated Deduction, CADE-24, Lake Placid, NY, USA, June 9-14, 2013*. Ed. by Maria Paola Bonacina. Vol. 7898. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2013, pp. 330–344. DOI: 10.1007/978-3-642-38574-2_23 (cit. on pp. 58, 84, 87, 88, 93, 94, 96, 97, 99, 117).

[BBL15a] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. "Temporal Conjunctive Queries in Expressive Description Logics with Transitive Roles". In: *Advances in Artificial Intelligence - 28th Australasian Joint Conference, AI 2015, Canberra, ACT, Australia, November 30-December 4, 2015*. Ed. by Bernhard Pfahringer and Jochen Renz. Vol. 9457. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2015, pp. 21–33. DOI: 10.1007/978-3-319-26350-2_3 (cit. on pp. 84, 85).

[BBL15b] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. "Temporal query entailment in the Description Logic SHQ". In: *Journal of Web Semantics* 33 (2015), pp. 71–93. ISSN: 1570-8268. DOI: 10.1016/j.websem.2014.11.008 (cit. on pp. 84, 96).

[BBL76] Barry W. Boehm, John R. Brown, and Myron Lipow. "Quantitative Evaluation of Software Quality". In: *Proceedings of the 2nd International Conference on Software Engineering, San Francisco, California, USA, October 13-15, 1976*. Ed. by Raymond T. Yeh and C. V. Ramamoorthy. IEEE Computer Society, 1976, pp. 592–605. URL: http://dl.acm.org/citation.cfm?id=807736 (cit. on p. 39).

[BBP17] Franz Baader, Stefan Borgwardt, and Rafael Peñaloza. "Decidability and Complexity of Fuzzy Description Logics". In: *Künstliche Intelligenz* 31.1 (2017), pp. 85–90. DOI: 10.1007/S13218-016-0459-3 (cit. on p. 80).

[Ber+22] Felix Beringhoff, Joel Greenyer, Christian Roesener, and Matthias Tichy. "Thirty-One Challenges in Testing Automated Vehicles: Interviews with Experts from Industry and Research". In: *2022 IEEE Intelligent Vehicles Symposium, IV 2022, Aachen, Germany, June 4-9, 2022*. New York, NY,

USA: IEEE, 2022, pp. 360–366. DOI: 10.1109/IV51971.2022.9827097 (cit. on p. 18).

[BGL12] Franz Baader, Silvio Ghilardi, and Carsten Lutz. "LTL over description logic axioms". In: *ACM Transactions on Computational Logic* 13.3 (2012), pp. 1–32. DOI: 10.1145/2287718.2287721 (cit. on p. 86).

[BHS08] Franz Baader, Ian Horrocks, and Ulrike Sattler. "Description Logics". In: *Handbook of Knowledge Representation*. Ed. by Frank van Harmelen, Vladimir Lifschitz, and Bruce W. Porter. Vol. 3. Foundations of Artificial Intelligence. Amsterdam, The Netherlands: Elsevier, 2008, pp. 135–179. ISBN: 978-0-444-52211-5. DOI: 10.1016/S1574-6526(07)03003-9 (cit. on p. 61).

[BK08] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. Cambridge, MA, USA: MIT Press, 2008. ISBN: 978-0-262-02649-9. URL: https://mitpress.mit.edu/9780262026499/principles-of-model-checking/ (cit. on p. 113).

[BLT13] Stefan Borgwardt, Marcel Lippmann, and Veronika Thost. "Temporal Query Answering in the Description Logic DL-Lite". In: *Frontiers of Combining Systems - 9th International Symposium, FroCoS 2013, Nancy, France, September 18-20, 2013*. Ed. by Pascal Fontaine, Christophe Ringeissen, and Renate A. Schmidt. Vol. 8152. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2013, pp. 165–180. DOI: 10.1007/978-3-642-40885-4_11 (cit. on pp. 85, 86).

[BLT15] Stefan Borgwardt, Marcel Lippmann, and Veronika Thost. "Temporalizing rewritable query languages over knowledge bases". In: *Journal of Web Semantics* 33 (2015), pp. 50–70. ISSN: 1570-8268. DOI: 10.1016/J.WEBSEM.2014.11.007 (cit. on p. 85).

[Boc+19] Florian Bock, Christoph Sippl, Aaron Heinz, Christoph Lauer, and Reinhard German. "Advantageous Usage of Textual Domain-Specific Languages for Scenario-Driven Development of Automated Driving Functions". In: *2019 IEEE International Systems Conference, SysCon 2019, Orlando, FL, USA, April 8-11, 2019*. New York, NY, USA: IEEE, 2019, pp. 1–8. DOI: 10.1109/SYSCON.2019.8836912 (cit. on p. 35).

[Boc+20] Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. "The inD Dataset: A Drone Dataset of Naturalistic Road User Trajectories at German Intersections". In: *IEEE Intelligent Vehicles Symposium, IV 2020, Las Vegas, NV, USA, October 19 - November 13, 2020*. New York, NY, USA: IEEE, 2020, pp. 1929–1934. DOI: 10.1109/IV47402.2020.9304839 (cit. on pp. 35, 145, 180).

[Böd+18] Eckard Böde, Matthias Büker, Ulrich Eberle, Martin Fränzle, Sebastian Gerwinn, and Birte Kramer. "Efficient Splitting of Test and Simulation Cases for the Verification of Highly Automated Driving Functions". In: *Computer Safety, Reliability, and Security - 37th International Conference, SAFECOMP 2018, Västerås, Sweden, September 19-21, 2018*. Ed. by Barbara Gallina, Amund Skavhaug, and Friedemann Bitsch. Vol. 11093. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2018, pp. 139–153. DOI: 10.1007/978-3-319-99130-6_10 (cit. on p. 37).

[Bor+25] Philipp Borchers, Tjark Koopmann, Lukas Westhofen, Jan Steffen Becker, Lina Putze, Dominik Grundt, Thies de Graaff, Vincent Kalwa, and Christian Neurohr. "TSC2CARLA: An abstract scenario-based verification toolchain for automated driving systems". In: *Science of Computer Programming* 242 (2025), p. 103256. DOI: 10.1016/J.SCICO.2024.103256 (cit. on p. 37).

[Bra+17] Sebastian Brandt, Elem Güzel Kalayci, Roman Kontchakov, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyaschev. "Ontology-Based Data Access with a Horn Fragment of Metric Temporal Logic". In: *The Thirty-First AAAI Conference on Artificial Intelligence, AAAI, 2017, The Twenty-Ninth Innovative Applications of Artificial Intelligence Conference, IAAI 2017, The Seventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2017, February 4-9, 2017, San Francisco, California, USA*. Ed. by Satinder Singh and Shaul Markovitch. Palo Alto, CA, USA: AAAI Press, 2017, pp. 1070–1076. DOI: 10.1609/AAAI.V31I1.10696 (cit. on p. 86).

[BT15] Stefan Borgwardt and Veronika Thost. "Temporal Query Answering in the Description Logic EL". In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. Ed. by Qiang Yang and Michael J. Wooldridge. Palo Alto, CA, USA: AAAI Press, 2015, pp. 2819–2825. URL: http://ijcai.org/Abstract/15/399 (cit. on p. 85).

[Bue+17] Martin Buechel, Gereon Hinz, Frederik Ruehl, Hans Schroth, Csaba Gyoeri, and Alois C. Knoll. "Ontology-based traffic scene modeling, traffic regulations dependent situational awareness and decision-making for automated vehicles". In: *IEEE Intelligent Vehicles Symposium, IV 2017, Los Angeles, CA, USA, June 11-14, 2017*. New York, NY, USA: IEEE, 2017, pp. 1471–1476. DOI: 10.1109/IVS.2017.7995917 (cit. on pp. 48, 51).

[Cal+07] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. "Tractable Reasoning and Efficient Query Answering in Description Logics: The *DL-Lite* Family". In: *Journal of Automated*

Reasoning 39.3 (2007), pp. 385–429. DOI: 10.1007/S10817-007-9078-X (cit. on p. 85).

[Cal+98] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Daniele Nardi, and Riccardo Rosati. "Description Logic Framework for Information Integration". In: *Principles of Knowledge Representation and Reasoning: Proceedings of the 6th International Conference, KR 1998, Trento, Italy, June 2-5, 1998*. Ed. by Anthony G. Cohn, Lenhart K. Schubert, and Stuart C. Shapiro. Burlington, MA, USA: Morgan Kaufmann, 1998, pp. 2–13. URL: https://dl.acm.org/doi/10.5555/3087454.3087456 (cit. on p. 3).

[Car79] William L. Carlson. "Crash injury prediction model". In: *Accident Analysis & Prevention* 11.2 (1979), pp. 137–153. ISSN: 0001-4575. DOI: https://doi.org/10.1016/0001-4575(79)90022-8 (cit. on p. 46).

[CGL08] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. "Conjunctive query containment and answering under description logic constraints". In: *ACM Transactions on Computational Logic* 9.3 (2008), 22:1–22:31. DOI: 10.1145/1352582.1352590 (cit. on p. 76).

[Cho90] Jan Chomicki. "Polynomial Time Query Processing in Temporal Deductive Databases". In: *Proceedings of the Ninth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS 1990, Nashville, TN, USA, April 2-4, 1990*. Ed. by Daniel J. Rosenkrantz and Yehoshua Sagiv. New York, NY, USA: ACM, 1990, pp. 379–391. DOI: 10.1145/298514.298589 (cit. on p. 86).

[CHY18] Chih-Hong Cheng, Chung-Hao Huang, and Hirotoshi Yasuoka. "Quantitative Projection Coverage for Testing ML-enabled Autonomous Systems". In: *Automated Technology for Verification and Analysis - 16th International Symposium, ATVA 2018, Los Angeles, CA, USA, October 7-10, 2018, Proceedings*. Ed. by Shuvendu K. Lahiri and Chao Wang. Vol. 11138. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2018, pp. 126–142. DOI: 10.1007/978-3-030-01090-4_8 (cit. on p. 163).

[CK24] Julia Chuzhoy and Sanjeev Khanna. "A Faster Combinatorial Algorithm for Maximum Bipartite Matching". In: *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*. Ed. by David P. Woodruff. SIAM, 2024, pp. 2185–2235. DOI: 10.1137/1.9781611977912.79 (cit. on p. 173).

[CKS81] Ashok K. Chandra, Dexter Kozen, and Larry J. Stockmeyer. "Alternation". In: *Journal of the ACM* 28.1 (1981), pp. 114–133. DOI: 10.1145/322234.322243 (cit. on p. 103).

[Cle00] P. L. Clemens. *System Safety Scrapbook*. 9th ed. Tullahoma, TN, USA: Sverdrup Technology, Inc., 2000. URL: https://isss-tvc.org/scrapbook.pdf (cit. on p. 26).

[Dam+17] Werner Damm, Stephanie Kemper, Eike Möhlmann, Thomas Peikenkamp, and Astrid Rakow. *Traffic Sequence Charts – From Visualization to Semantics*. Technical Report. Oldenburg, Germany: SFB/TR 14 AVACS, Oct. 2017. DOI: 10.13140/RG.2.2.15190.42563 (cit. on p. 35).

[Dar+12] Raghad Dardar, Barbara Gallina, Andreas Johnsen, Kristina Lundqvist, and Mattias Nyberg. "Industrial Experiences of Building a Safety Case in Compliance with ISO 26262". In: *23rd IEEE International Symposium on Software Reliability Engineering Workshops, ISSRE 2012 Workshops, Dallas, TX, USA, November 27-30, 2012*. Washington, D.C., USA: IEEE Computer Society, 2012, pp. 349–354. DOI: 10.1109/ISSREW.2012.86 (cit. on p. 18).

[Dig24] *Amtsblatt des Bundesministeriums für Digitales und Verkehr der Bundesrepublik Deutschland (VkBl.)* 78.3 (2024). Ausgegeben zu Bonn am 15. Februar 2024. URL: https://fragdenstaat.de/dokumente/255595-vkbl-nr-3-2024/ (cit. on p. 149).

[Din+06] Thomas A. Dingus, Sheila G. Klauer, Vicki Lewis Neale, Andy Petersen, Suzanne E. Lee, Jeremy Sudweeks, Miguel A. Perez, Jonathan Hankey, David Ramsey, Santosh Gupta, et al. *The 100-car naturalistic driving study, Phase II-results of the 100-car field experiment*. Technical Report. Washington, D.C., USA, 2006. URL: https://rosap.ntl.bts.gov/view/dot/37370 (cit. on p. 43).

[Dos+17] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. "CARLA: An Open Urban Driving Simulator". In: *Proceedings of the 1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017*. Cambridge, UK: ML Research Press, Nov. 2017, pp. 1–16. URL: https://proceedings.mlr.press/v78/dosovitskiy17a.html (cit. on p. 36).

[Els+20] Philip Elspas, Jacob Langner, Michael Aydinbas, Johannes Bach, and Eric Sax. "Leveraging Regular Expressions for Flexible Scenario Detection in Recorded Driving Data". In: *IEEE International Symposium on Systems Engineering, ISSE 2020, Vienna, Austria, October 12 - November 12, 2020*. New York, NY, USA: IEEE, 2020, pp. 1–8. DOI: 10.1109/ISSE49799.2020.9272025 (cit. on pp. 6, 48, 49).

[ES13] Karim El-Basyouny and Tarek Sayed. "Safety performance functions using traffic conflicts". In: *Safety science* 51.1 (2013), pp. 160–164. DOI: 10.1016/j.ssci.2012.04.015 (cit. on p. 43).

[Est22] Klemens Esterle. "Formalizing and Modeling Traffic Rules Within Interactive Behavior Planning". PhD thesis. München, Germany: Technische Universität München, 2022. URL: https://nbn-resolving.org/urn:nbn:de:bvb:91-diss-20211209-1601896-1-3 (cit. on pp. 48, 50).

[Fab00] Francesco M. Donini and Fabio Massacci. "EXPTIME tableaux for ALC". In: *Artificial Intelligence* 124.1 (2000), pp. 87–138. DOI: 10.1016/S0004-3702(00)00070-9 (cit. on p. 66).

[Fav20] Marco Favorito. "Standard Grammars for LTL and LDL (v0.1.0)". In: *CoRR* abs/2012.13638 (2020). DOI: 10.48550/arXiv.2012.13638 (cit. on p. 138).

[GA21] Luis Gressenbuch and Matthias Althoff. "Predictive Monitoring of Traffic Rules". In: *24th IEEE International Intelligent Transportation Systems Conference, ITSC 2021, Indianapolis, IN, USA, September 19-22, 2021*. New York, NY, USA: IEEE, 2021, pp. 915–922. DOI: 10.1109/ITSC48978.2021.9564432 (cit. on pp. 48, 50).

[Gay+14] Gregory Gay, Matt Staats, Michael W. Whalen, and Mats Per Erik Heimdahl. "Moving the goalposts: coverage satisfaction is not enough". In: *7th International Workshop on Search-Based Software Testing, SBST 2014, Hyderabad, India, June 2, 2014*. Ed. by Phil McMinn and Mark Harman. New York, NY, USA: ACM, 2014, pp. 19–22. DOI: 10.1145/2593833.2593837 (cit. on p. 163).

[GBO24] Erwin de Gelder, Maren Buermann, and Olaf Op den Camp. "Coverage Metrics for a Scenario Database for the Scenario-Based Assessment of Automated Driving Systems". In: *IEEE International Automated Vehicle Validation Conference, IAVVC 2024, Pittsburgh, PA, USA, October 21-23, 2024*. New York, NY, USA: IEEE, 2024, pp. 1–8. DOI: 10.1109/IAVVC63304.2024.10786405 (cit. on p. 163).

[Gel+20] Erwin de Gelder, Jeroen Manders, Corrado Grappiolo, Jan-Pieter Paardekooper, Olaf Op den Camp, and Bart De Schutter. "Real-World Scenario Mining for the Assessment of Automated Vehicles". In: *23rd IEEE International Conference on Intelligent Transportation Systems, ITSC 2020, Rhodes, Greece, September 20-23, 2020*. New York, NY, USA: IEEE, 2020, pp. 1–8. DOI: 10.1109/ITSC45102.2020.9294652 (cit. on pp. 6, 48, 49).

[Gel+22] Erwin de Gelder, Jan-Pieter Paardekooper, Arash Khabbaz Saberi, Hala Elrofai, Olaf Op den Camp, Steven B. Kraines, Jeroen Ploeg, and Bart De Schutter. "Towards an Ontology for Scenario Definition for the Assessment of Automated Vehicles: An Object-Oriented Framework". In: *IEEE Transactions on Intelligent Vehicles* 7.2 (2022), pp. 300–314. DOI: 10.1109/TIV.2022.3144803 (cit. on pp. 48, 49).

[Gel+25] Erwin de Gelder, Tajinder Singh, Hadj-Selem Fouad, Sergi Vidal Bazan, and Olaf Op den Camp. "Scenario Metrics for the Safety Assurance Framework of Automated Vehicles: A Review of Its Application". In: *Vehicles* 7.3 (2025), p. 100. DOI: 10.3390/vehicles7030100 (cit. on p. 163).

[GF21] Giuseppe De Giacomo and Marco Favorito. "Compositional Approach to Translate LTLf/LDLf into Deterministic Finite Automata". In: *Proceedings of the Thirty-First International Conference on Automated Planning and Scheduling, ICAPS 2021, Guangzhou, China (virtual), August 2-13, 2021*. Ed. by Susanne Biundo, Minh Do, Robert Goldman, Michael Katz, Qiang Yang, and Hankz Hankui Zhuo. Palo Alto, USA: AAAI Press, 2021, pp. 122–130. URL: https://ojs.aaai.org/index.php/ICAPS/article/view/15954 (cit. on pp. 103, 126, 140).

[GJK16] Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Roman Kontchakov. "Temporalized EL Ontologies for Accessing Temporal Data: Complexity of Atomic Queries". In: *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, July 9-15 2016*. Ed. by Subbarao Kambhampati. Palo Alto, CA, USA: IJCAI/AAAI Press, 2016, pp. 1102–1108. URL: http://www.ijcai.org/Abstract/16/160 (cit. on p. 87).

[GJO16] Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Ana Ozaki. "On Metric Temporal Description Logics". In: *22nd European Conference on Artificial Intelligence, ECAI 2016, Pitesti, Romania, July 1-3, 2021*. Ed. by Gal A. Kaminka, Maria Fox, Paolo Bouquet, Eyke Hüllermeier, Virginia Dignum, Frank Dignum, and Frank van Harmelen. Vol. 285. Frontiers in Artificial Intelligence and Applications. Amsterdam, The Netherlands: IOS Press, 2016, pp. 837–845. DOI: 10.3233/978-1-61499-672-9-837 (cit. on p. 87).

[GK12] Víctor Gutiérrez-Basulto and Szymon Klarman. "Towards a Unifying Approach to Representing and Querying Temporal Data in Description Logics". In: *Web Reasoning and Rule Systems - 6th International Conference, RR 2012, Vienna, Austria, September 10-12, 2012*. Ed. by Markus Krötzsch and Umberto Straccia. Vol. 7497. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2012, pp. 90–105. DOI: 10.1007/978-3-642-33203-6_8 (cit. on pp. 7, 84).

[Gli+08] Birte Glimm, Carsten Lutz, Ian Horrocks, and Ulrike Sattler. "Conjunctive Query Answering for the Description Logic SHIQ". In: *Journal of Artificial Intelligence Research* 31 (2008), pp. 157–204. DOI: 10.1613/JAIR.2372 (cit. on pp. 61, 76, 77, 94, 108).

[Gmb16] VIRES Simulationstechnologie GmbH. *OpenSCENARIO – Project*. Accessed on 10th of September 2025. Feb. 2016. URL: https://web.archive.org/web/20160220045425/http://openscenario.org/project.html (cit. on p. 35).

[GMG85] Lorne Greenspan, Barry A McLellan, and Helen Greig. "Abbreviated injury scale and injury severity score: a scoring chart". In: *Journal of Trauma and Acute Care Surgery* 25.1 (1985), pp. 60–64. DOI: 10.1097/00005373-198501000-00010 (cit. on p. 30).

[GPH05] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. "LUBM: A benchmark for OWL knowledge base systems". In: *Journal of Web Semantics* 3.2 (2005), pp. 158–182. ISSN: 1570-8268. DOI: 10.1016/j.websem.2005.06.005 (cit. on p. 145).

[GV13] Giuseppe De Giacomo and Moshe Y. Vardi. "Linear Temporal Logic and Linear Dynamic Logic on Finite Traces". In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013, Beijing, China, August 3-9, 2013*. Ed. by Francesca Rossi. Palo Alto, CA, USA: IJCAI/AAAI Press, 2013, pp. 854–860. URL: http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6997 (cit. on pp. 50, 90, 98, 103, 106, 107, 121, 132).

[Hay72] John C. Hayward. "Near miss determination through use of a scale of danger". In: *51st Annual Meeting of the Highway Research Board, Washington, D.C., USA, January 17-21, 1972*. Vol. 384. Highway Research Record. Washington, D.C., USA: Highway Research Board, 1972, pp. 24–34. URL: http://onlinepubs.trb.org/Onlinepubs/hrr/1972/384/384-004.pdf (cit. on pp. 40, 43).

[Hei+19] Robin Heinzler, Philipp Schindler, Jürgen Seekircher, Werner Ritter, and Wilhelm Stork. "Weather Influence and Classification with Automotive Lidar Sensors". In: *2019 IEEE Intelligent Vehicles Symposium, IV 2019, Paris, France, June 9-12, 2019*. New York, NY, USA: IEEE, 2019, pp. 1527–1534. DOI: 10.1109/IVS.2019.8814205 (cit. on p. 38).

[Hil+11] Martin Hilscher, Sven Linker, Ernst-Rüdiger Olderog, and Anders P. Ravn. "An Abstract Model for Proving Safety of Multi-lane Traffic Manoeuvres". In: *Formal Methods and Software Engineering - 13th International Conference on Formal Engineering Methods, ICFEM 2011, Durham, UK, October 26-28, 2011*. Ed. by Shengchao Qin and Zongyan Qiu. Vol. 6991. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2011, pp. 404–419. DOI: 10.1007/978-3-642-24559-6_28 (cit. on p. 35).

[HK71] John E. Hopcroft and Richard M. Karp. "A n^5/2 Algorithm for Maximum Matchings in Bipartite Graphs". In: *12th Annual Symposium on Switching and Automata Theory, East Lansing, Michigan, USA, October 13-15, 1971*. IEEE Computer Society, 1971, pp. 122–125. DOI: 10.1109/SWAT.1971.1 (cit. on p. 173).

[HKS06] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. "The Even More Irresistible SROIQ". In: *Principles of Knowledge Representation and Reasoning: Proceedings of the 10th International Conference, KR 2006, Lake District, UK, June 2-5, 2006*. Ed. by Patrick Doherty, John Mylopoulos, and Christopher A. Welty. Palo Alto, USA: AAAI Press, 2006, pp. 57–67. URL: http://www.aaai.org/Library/KR/2006/kr06-009.php (cit. on pp. 68, 70, 71).

[HMT01] Volker Haarslev, Ralf Möller, and Anni-Yasmin Turhan. "Exploiting Pseudo Models for TBox and ABox Reasoning in Expressive Description Logics". In: *Automated Reasoning, 1st International Joint Conference, IJCAR 2001, Siena, Italy, June 18-23, 2001*. Ed. by Rajeev Goré, Alexander Leitsch, and Tobias Nipkow. Vol. 2083. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2001, pp. 61–75. DOI: 10.1007/3-540-45744-5_6 (cit. on p. 82).

[HMU07] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to automata theory, languages, and computation*. 3rd ed. Pearson New International Edition. London, UK: Pearson, 2007. ISBN: 978-0-321-47617-3. URL: https://www.pearson.com/en-us/subject-catalog/p/introduction-to-automata-theory-languages-and-computation/P200000003517/9780321455369 (cit. on p. 59).

[HS01] Ian Horrocks and Ulrike Sattler. "Ontology Reasoning in the SHOQ(D) Description Logic". In: *Proceedings of the 17th International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001*. Ed. by Bernhard Nebel. Burlington, MA, USA: Morgan Kaufmann, 2001, pp. 199–204. URL: https://dl.acm.org/doi/10.5555/1642090.1642117 (cit. on pp. 68, 70, 71).

[HS04] Ian Horrocks and Ulrike Sattler. "Decidability of SHIQ with complex role inclusion axioms". In: *Artificial Intelligence* 160.1-2 (2004), pp. 79–104. DOI: 10.1016/J.ARTINT.2004.06.002 (cit. on p. 69).

[HSK06] Jörg Hillenbrand, Andreas M. Spieker, and Kristian Kroschel. "A multilevel collision mitigation approach–Its situation assessment, decision making, and performance tradeoffs". In: *IEEE Transactions on Intelligent Transportation Systems* 7.4 (2006), pp. 528–540. DOI: 10.1109/TITS.2006.883115 (cit. on pp. 46, 47).

[HT00] Ian Horrocks and Sergio Tessaris. "A Conjunctive Query Language for Description Logic Aboxes". In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, Austin, Texas, USA, July 30-August 3, 2000*. Ed. by Henry A. Kautz and Bruce W. Porter. Palo Alto, CA, USA: AAAI Press / The MIT Press, 2000, pp. 399–404. URL: http://www.aaai.org/Library/AAAI/2000/aaai00-061.php (cit. on pp. 3, 76–80, 105, 140).

[HT90] Richard G. Hamlet and Ross Taylor. "Partition Testing Does Not Inspire Confidence". In: *IEEE Transactions on Software Engineering* 16.12 (1990), pp. 1402–1411. DOI: 10.1109/32.62448 (cit. on p. 163).

[Hua+16] Wuling Huang, Kunfeng Wang, Yisheng Lv, and Fenghua Zhu. "Autonomous vehicles testing methods review". In: *19th IEEE International Conference on Intelligent Transportation Systems, ITSC 2016, Rio de Janeiro, Brazil, November 1-4, 2016*. New York, NY, USA: IEEE, 2016, pp. 163–168. DOI: 10.1109/ITSC.2016.7795548 (cit. on p. 37).

[Hül+10] Till Hülnhagen, Ingo Dengler, Andreas Tamke, Thao Dang, and Gabi Breuel. "Maneuver recognition using probabilistic finite-state machines and fuzzy logic". In: *IEEE Intelligent Vehicles Symposium, 2010, La Jolla, CA, USA, June 21-24, 2010*. New York, NY, USA: IEEE, 2010, pp. 65–70. DOI: 10.1109/IVS.2010.5548066 (cit. on pp. 6, 48, 49).

[Hum09] Britta Hummel. "Description logic for scene understanding at the example of urban road intersections". PhD thesis. Karlsruhe, Germany: Universität Karlsruhe (TH), 2009. ISBN: 3-8381-1423-X. DOI: 10.5445/IR/1000015554 (cit. on pp. 48, 51).

[Hyd75] Christer Hydén. *Relations between serious conflicts and traffic accidents*. Technical Report. Lund, Sweden: Tekniska Högskolan i Lund, Institutionen för Trafikteknik, 1975. URL: https://trid.trb.org/View/43563 (cit. on p. 40).

[HZW11] Michael Hülsen, Johann Marius Zöllner, and Christian Weiss. "Traffic intersection situation description ontology for advanced driver assistance". In: *IEEE Intelligent Vehicles Symposium, IV 2011, Baden-Baden, Germany, June 5-9, 2011*. New York, NY, USA: IEEE, 2011, pp. 993–999. DOI: 10.1109/IVS.2011.5940415 (cit. on pp. 48, 51).

[IHS15] Malte Isberner, Falk Howar, and Bernhard Steffen. "The Open-Source LearnLib - A Framework for Active Automata Learning". In: *Computer Aided Verification - 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings Part I*. Ed. by Daniel Kroening

and Corina S. Pasareanu. Vol. 9206. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2015, pp. 487–495. DOI: 10.1007/978-3-319-21690-4_32 (cit. on p. 140).

[Int14] International Organization for Standardization. *ISO/IEC Guide 51: Safety aspects — Guidelines for their inclusion in standards*. Standard. Geneva, Switzerland, 2014. URL: https://www.iso.org/standard/53940.html (cit. on pp. 23, 24).

[Int18] International Organization for Standardization. *ISO 26262: Road vehicles – Functional safety*. Standard. Geneva, Switzerland, 2018. URL: https://www.iso.org/standard/68383.html (cit. on pp. 18–21, 34).

[Int22] International Organization for Standardization. *ISO 21448: Road vehicles – Safety of the intended functionality*. Standard. Geneva, Switzerland, 2022. URL: https://www.iso.org/standard/77490.html (cit. on pp. 2, 5, 17, 20–22, 24, 26–28, 30, 34).

[Jan05] Jonas Jansson. "Collision Avoidance Theory: With application to automotive collision mitigation". PhD thesis. Linköping, Sweden: Linköping University, 2005. ISBN: 91-85299-45-6. URL: https://liu.diva-portal.org/smash/record.jsf?pid=diva:617438 (cit. on pp. 45, 46).

[JLD18] Carl Johnsson, Aliaksei Laureshyn, and Tim De Ceunynck. "In search of surrogate safety indicators for vulnerable road users: a review of surrogate safety indicators". In: *Transport Reviews* 38.6 (2018), pp. 765–785. DOI: 10.1080/01441647.2018.1442888 (cit. on pp. 43–45).

[JLD21] Carl Johnsson, Aliaksei Laureshyn, and Carmelo Dágostino. "A relative approach to the validation of surrogate measures of safety". In: *Accident Analysis & Prevention* 161 (2021), p. 106350. DOI: 10.1016/j.aap.2021.106350 (cit. on pp. 43, 45).

[Jok93] H. C. Joksch. "Velocity change and fatality risk in a crash–a rule of thumb". In: *Accident Analysis & Prevention* 25 (1 1993), pp. 103–104. DOI: 10.1016/0001-4575(93)90102-3 (cit. on p. 46).

[JR93] Tao Jiang and Bala Ravikumar. "Minimal NFA Problems are Hard". In: *SIAM Journal on Computing* 22.6 (1993), pp. 1117–1141. DOI: 10.1137/0222067 (cit. on p. 103).

[Kal+18] Elem Güzel Kalayci, Guohui Xiao, Vladislav Ryzhikov, Tahir Emre Kalayci, and Diego Calvanese. "Ontop-temporal: A Tool for Ontology-based Query Answering over Temporal Data". In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*. Ed. by Alfredo Cuzzocrea, James Allan,

Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang. New York, NY, USA: ACM, 2018, pp. 1927–1930. DOI: 10.1145/3269206.3269230 (cit. on pp. 3, 86).

[Kaz08] Yevgeny Kazakov. "RIQ and SROIQ Are Harder than SHOIQ". In: *Principles of Knowledge Representation and Reasoning: Proceedings of the 11th International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*. Ed. by Gerhard Brewka and Jérôme Lang. Palo Alto, CA, USA: AAAI Press, 2008, pp. 274–284. URL: http://www.aaai.org/Library/KR/2008/kr08-027.php (cit. on p. 72).

[KD20] Abolfazl Karimi and Parasara Sridhar Duggirala. "Formalizing traffic rules for uncontrolled intersections". In: *11th ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS 2020, Sydney, Australia, April 21-25, 2020*. New York, NY, USA: IEEE, 2020, pp. 41–50. DOI: 10.1109/ICCPS48487.2020.00012 (cit. on pp. 48, 51).

[KE13] Stephanie Kemper and Christoph Etzien. "A Visual Logic for the Description of Highway Traffic Scenarios". In: *Complex Systems Design & Management, Proceedings of the Fourth International Conference on Complex Systems Design & Management CSD&M 2013, Paris, France, December 4-6, 2013*. Ed. by Marc Aiguier, Frédéric Boulanger, Daniel Krob, and Clotilde Marchal. Berlin, Germany: Springer, 2013, pp. 233–245. DOI: 10.1007/978-3-319-02812-5_17 (cit. on p. 35).

[Kea+99] Joseph Kearney, Peter Willemsen, Stéphane Donikian, and Frédéric Devillers. "Scenario languages for driving simulation". In: *Proceedings of the Driving Simulation Conference 1999, DSC'99, Paris, France, July, 1999*. Boulogne-Billancourt, France: Driving Simulation Association, 1999. URL: https://www.researchgate.net/publication/228549489_Scenario_languages_for_driving_simulation (cit. on p. 35).

[KH24] Michael Kochem and Stephan Hakuli. "Virtuelle Integration". In: *Handbuch Assistiertes und Automatisiertes Fahren*. Ed. by Hermann Winner, Klaus C. J. Dietmayer, Lutz Eckstein, Meike Jipp, Markus Maurer, and Christoph Stiller. Berlin, Germany: Springer, 2024, pp. 131–152. ISBN: 978-3-658-38486-9. DOI: 10.1007/978-3-658-38486-9_7 (cit. on p. 37).

[Kha+17] Evgeny Kharlamov, Dag Hovland, Martin G. Skjæveland, Dimitris Bilidas, Ernesto Jiménez-Ruiz, Guohui Xiao, Ahmet Soylu, Davide Lanti, Martin Rezk, Dmitriy Zheleznyakov, Martin Giese, Hallstein Lie, Yannis E. Ioannidis, Yannis Kotidis, Manolis Koubarakis, and Arild Waaler. "Ontology Based

Data Access in Statoil". In: *Journal of Web Semantics* 44 (2017), pp. 3–36. DOI: 10.1016/J.WEBSEM.2017.05.005 (cit. on p. 3).

[KK11] Petr Kremen and Zdenek Kouba. "Conjunctive Query Optimization in OWL2-DL". In: *Database and Expert Systems Applications - 22nd International Conference, DEXA 2011, Toulouse, France, August 29-September 2, 2011, Proceedings Part II*. Ed. by Abdelkader Hameurlain, Stephen W. Liddle, Klaus-Dieter Schewe, and Xiaofang Zhou. Vol. 6861. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2011, pp. 188–202. DOI: 10.1007/978-3-642-23091-2_18 (cit. on pp. 81, 83).

[KMS02] Nils Klarlund, Anders Møller, and Michael I. Schwartzbach. "MONA Implementation Secrets". In: *International Journal of Foundations of Computer Science* 13.4 (2002), pp. 571–586. DOI: 10.1142/S012905410200128X (cit. on p. 103).

[Koo+25] Tjark Koopmann, Lina Putze, Lukas Westhofen, Roman Gansch, Ahmad Adee, and Christian Neurohr. "Grasping Causality for the Explanation of Criticality for Automated Driving". In: *IEEE Access* 13 (2025), pp. 54739–54756. DOI: 10.1109/ACCESS.2025.3555177 (cit. on p. 149).

[KOW19] Philip Koopman, Beth Osyk, and Jack Weast. "Autonomous Vehicles Meet the Physical World: RSS, Variability, Uncertainty, and Proving Safety". In: *Computer Safety, Reliability, and Security - 38th International Conference, SAFECOMP 2019, Turku, Finland, September 11-13, 2019*. Ed. by Alexander B. Romanovsky, Elena Troubitsyna, and Friedemann Bitsch. Vol. 11698. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2019, pp. 245–253. DOI: 10.1007/978-3-030-26601-1_17 (cit. on p. 44).

[Koy90] Ron Koymans. "Specifying Real-Time Properties with Metric Temporal Logic". In: *Real-Time Systems* 2.4 (1990), pp. 255–299. DOI: 10.1007/BF01995674 (cit. on p. 90).

[KP16] Nidhi Kalra and Susan M Paddock. "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" In: *Transportation Research Part A: Policy and Practice* 94 (2016), pp. 182–193. DOI: 10.1016/j.tra.2016.09.010 (cit. on pp. 18, 29, 34).

[Kra+02] Daniel Krajzewicz, Georg Hertkorn, Christian Rössel, and Peter Wagner. "SUMO (Simulation of Urban MObility) – an open-source traffic simulation". In: *Proceedings of the 4th Middle East Symposium on Simulation and Modelling, MESM 2002, Sharjah, UAE, September 28-30, 2002*. Ostend, Belgium: The European Multidisciplinary Society for Modelling and Simulation Technology, 2002, pp. 183–187. URL: https://eurosis.org/cms/files/proceedings_full/MESM20021page.pdf (cit. on p. 36).

[Kre88] Mark W. Krentel. "The Complexity of Optimization Problems". In: *Journal of Computer and System Sciences* 36.3 (1988), pp. 490–509. DOI: 10.1016/0022-0000(88)90039-6 (cit. on p. 172).

[KRH13] Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. "Complexities of Horn Description Logics". In: *ACM Transactions on Computational Logic* 14.1 (2013), 2:1–2:36. DOI: 10.1145/2422085.2422087 (cit. on p. 121).

[Kru+05] Hans-Peter Krueger, Martin Grein, Armin Kaussner, Christian Mark, et al. "SILAB – A task-oriented driving simulation". In: *Proceedings of the Driving Simulation Conference, North America, DSC-NA 2005, Orlando, FL, USA, November 30-December 2, 2005*. Iowa City, Iowa, USA: Driving Safety Research Institute, University of Iowa, 2005, pp. 323–331. URL: http://www.nads-sc.uiowa.edu/dscna/2005/papers/SILAB_A_task_oriented_driving_simulation.pdf (cit. on p. 35).

[Kus+25] Kristofer D. Kusano, John M. Scanlon, Yin-Hsiu Chen, Timothy L. McMurry, Tilia Gode, and Trent Victor and. "Comparison of Waymo Rider-Only crash rates by crash type to human benchmarks at 56.7 million miles". In: *Traffic Injury Prevention* 0.0 (2025), pp. 1–13. DOI: 10.1080/15389588.2025.2499887 (cit. on pp. 2, 36).

[LA23] Yuanfei Lin and Matthias Althoff. "CommonRoad-CriMe: A Toolbox for Criticality Measures of Autonomous Vehicles". In: *IEEE Intelligent Vehicles Symposium, IV 2023, Anchorage, AK, USA, June 4-7, 2023*. New York, NY, USA: IEEE, 2023, pp. 1–8. DOI: 10.1109/IV55152.2023.10186673 (cit. on p. 52).

[Lam17] Jean-Baptiste Lamy. "Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies". In: *Artificial Intelligence in Medicine* 80 (2017), pp. 11–28. DOI: 10.1016/J.ARTMED.2017.07.002 (cit. on p. 150).

[Lin+22] Clemens Linnhoff, Kristof Hofrichter, Lukas Elster, Philipp Rosenberger, and Hermann Winner. "Measuring the Influence of Environmental Conditions on Automotive Lidar Sensors". In: *Sensors* 22.14 (2022), p. 5266. DOI: 10.3390/S22145266 (cit. on p. 38).

[Lip14] Marcel Lippmann. "Temporalised description logics for monitoring partially observable events". PhD thesis. Dresden, Germany: Technische Universität Dresden, 2014. URL: https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa-147977 (cit. on pp. 3, 7, 8, 84, 100, 101).

[Liu+21] Shuang Liu, Xuesong Wang, Omar Hassanin, Xiaoyan Xu, Minming Yang, David Hurwitz, and Xiangbin Wu. "Calibration and evaluation of responsibility-sensitive safety (RSS) in automated vehicle performance during cut-in scenarios". In: *Transportation Research Part C: Emerging Technologies* 125 (2021), p. 103037. DOI: 10.1016/j.trc.2021.103037 (cit. on p. 44).

[LÖ18] Ling Liu and M. Tamer Özsu, eds. *Encyclopedia of Database Systems*. 2nd ed. Berlin, Germany: Springer, 2018. ISBN: 978-1-4614-8266-6. DOI: 10.1007/978-1-4614-8265-9 (cit. on p. 53).

[Lor96] Dominique Lord. "Analysis of pedestrian conflicts with left-turning traffic". In: *Transportation Research Record* 1538.1 (1996), pp. 61–67. DOI: 10.1177/0361198196153800108 (cit. on p. 45).

[Ltd20] Foretellix Ltd. *Measurable Scenario Description Language Reference*. Accessed on 15th of April 2025. Ramat Gan, Israel, July 2020. URL: https://www.foretellix.com/wp-content/uploads/2020/07/M-SDL_LRM_OS.pdf (cit. on p. 35).

[Lu+21] Chang Lu, Xiaolin He, Hans van Lint, Huizhao Tu, Riender Happee, and Meng Wang. "Performance evaluation of surrogate measures of safety with naturalistic driving data". In: *Accident Analysis & Prevention* 162 (2021), p. 106403. DOI: 10.1016/j.aap.2021.106403 (cit. on p. 44).

[Luc+16] Alberto Lucchetti, Carlo Ongini, Simone Formentin, Sergio M. Savaresi, and Luigi Del Re. "Automatic recognition of driving scenarios for ADAS design". In: vol. 49. 11. 8th IFAC Symposium on Advances in Automotive Control, AAC 2016, Norrköping, Sweden, June 20-23, 2016. Amsterdam, The Netherlands: Elsevier, 2016, pp. 109–114. DOI: 10.1016/j.ifacol.2016.08.017 (cit. on pp. 6, 48, 49).

[Lüt+25] Florian Lüttner, Daniel Schmidt, Henning Höpfner, Frank Hantschel, Mirjam Fehling-Kaschek, Corinna Köpke, Ivo Häring, and Alexander Stolz. "Assessing Critical Traffic Scenarios: A Modular Evaluation Framework". In: *IEEE Transactions on Intelligent Transportation Systems* 26.6 (2025), pp. 8147–8161. DOI: 10.1109/TITS.2025.3546081 (cit. on pp. 45, 52).

[Lut07] Carsten Lutz. "Inverse Roles Make Conjunctive Queries Hard". In: *Proceedings of the 2007 International Workshop on Description Logics, DL 2007, Brixen-Bressanone, Bozen-Bolzano, Italy, June 8-10, 2007*. Ed. by Diego Calvanese, Enrico Franconi, Volker Haarslev, Domenico Lembo, Boris Motik, Anni-Yasmin Turhan, and Sergio Tessaris. Vol. 250. CEUR Workshop Proceedings. Aachen, Germany: CEUR-WS.org, 2007. URL: https://ceur-ws.org/Vol-250/paper_3.pdf (cit. on pp. 94, 120).

[Lut08] Carsten Lutz. "The Complexity of Conjunctive Query Answering in Expressive Description Logics". In: *Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008*. Ed. by Alessandro Armando, Peter Baumgartner, and Gilles Dowek. Vol. 5195. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2008, pp. 179–193. DOI: 10.1007/978-3-540-71070-7_16 (cit. on p. 77).

[LVR19] Jianwen Li, Moshe Y. Vardi, and Kristin Y. Rozier. "Satisfiability Checking for Mission-Time LTL". In: *Computer Aided Verification - 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings Part II*. Ed. by Isil Dillig and Serdar Tasiran. Vol. 11562. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2019, pp. 3–22. DOI: 10.1007/978-3-030-25543-5_1 (cit. on pp. 89, 92).

[LWZ08] Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. "Temporal Description Logics: A Survey". In: *15th International Symposium on Temporal Representation and Reasoning, TIME 2008, Montréal, Canada, June 16-18 2008*. Ed. by Stéphane Demri and Christian S. Jensen. Washington D.C., USA: IEEE Computer Society, 2008, pp. 3–14. DOI: 10.1109/TIME.2008.14 (cit. on pp. 86, 87).

[Mac+22] Steven Macenski, Tully Foote, Brian P. Gerkey, Chris Lalancette, and William Woodall. "Robot Operating System 2: Design, architecture, and uses in the wild". In: *Science Robotics* 7.66 (2022). DOI: 10.1126/SCIROBOTICS.ABM6074 (cit. on p. 38).

[Mah+17] SM Sohel Mahmud, Luis Ferreira, Md Shamsul Hoque, and Ahmad Tavassoli. "Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs". In: *IATSS Research* 41.4 (2017), pp. 153–163. DOI: 10.1016/j.iatssr.2017.02.001 (cit. on p. 40).

[Mai+18] David Maier, K. Tuncay Tekle, Michael Kifer, and David Scott Warren. "Datalog: concepts, history, and outlook". In: *Declarative Logic Programming: Theory, Systems, and Applications*. Ed. by Michael Kifer and Yanhong Annie Liu. Vol. 20. ACM Books. New York, NY, USA: ACM / Morgan & Claypool, 2018, pp. 3–100. ISBN: 978-1-970001-99-0. DOI: 10.1145/3191315.3191317 (cit. on p. 86).

[Mai+20] Sebastian Maierhofer, Anna-Katharina Rettinger, Eva Charlotte Mayer, and Matthias Althoff. "Formalization of Interstate Traffic Rules in Temporal Logic". In: *IEEE Intelligent Vehicles Symposium, IV 2020, Las Vegas, NV, USA, October 19 - November 13, 2020*. New York, NY, USA: IEEE, 2020, pp. 752–759. DOI: 10.1109/IV47402.2020.9304549 (cit. on pp. 48, 49).

[Mar+25] Victor M. Garcia Martinez, Salim S. Mussi, Moisés R. N. Ribeiro, Divanilson R. Campelo, and Tereza Cristina M. B. Carvalho. "VRU Safety at Road Intersections: An Ontology-Driven Digital Twin". In: *IEEE Access* 13 (2025), pp. 138166–138184. DOI: 10.1109/ACCESS.2025.3595998 (cit. on p. 148).

[Mat+22] André de Matos Pedro, Tomás Silva, Tiago F. Sequeira, João Lourenço, João Costa Seco, and Carla Ferreira. "Monitoring of Spatio-Temporal Properties with Nonlinear SAT Solvers". In: *Formal Methods for Industrial Critical Systems - 27th International Conference, FMICS 2022, Warsaw, Poland, September 14-15, 2022*. Ed. by Jan Friso Groote and Marieke Huisman. Vol. 13487. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2022, pp. 155–171. DOI: 10.1007/978-3-031-15008-1_11 (cit. on pp. 48, 49).

[MB08] Leonardo Mendonça de Moura and Nikolaj S. Bjørner. "Z3: An Efficient SMT Solver". In: *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008*. Ed. by C. R. Ramakrishnan and Jakob Rehof. Vol. 4963. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2008, pp. 337–340. DOI: 10.1007/978-3-540-78800-3_24 (cit. on pp. 170, 176).

[MBM18] Till Menzel, Gerrit Bagschik, and Markus Maurer. "Scenarios for Development, Test and Validation of Automated Vehicles". In: *2018 IEEE Intelligent Vehicles Symposium, IV 2018, Changshu, Suzhou, China, June 26-30, 2018*. New York, NY, USA: IEEE, 2018, pp. 1821–1827. DOI: 10.1109/IVS.2018.8500406 (cit. on p. 36).

[MHW06] Ralf Möller, Volker Haarslev, and Michael Wessel. "On the Scalability of Description Logic Instance Retrieval". In: *Proceedings of the 2006 International Workshop on Description Logics, DL 2006, Windermere, Lake District, UK, May 30-June 1, 2006*. Ed. by Bijan Parsia, Ulrike Sattler, and David Toman. Vol. 189. CEUR Workshop Proceedings. Aachen, Germany: CEUR-WS.org, 2006. URL: https://ceur-ws.org/Vol-189/submission_10.pdf (cit. on p. 82).

[Mic+20] Dimitrios Michail, Joris Kinable, Barak Naveh, and John V. Sichi. "JGraphT - A Java Library for Graph Data Structures and Algorithms". In: *ACM Transactions on Mathematical Software* 46.2 (2020), 16:1–16:29. DOI: 10.1145/3381449 (cit. on p. 176).

[MMA22] Sebastian Maierhofer, Paul Moosbrugger, and Matthias Althoff. "Formalization of Intersection Traffic Rules in Temporal Logic". In: *2022 IEEE Intelligent Vehicles Symposium, IV 2022, Aachen, Germany, June 4-9, 2022*. New York, NY, USA: IEEE, 2022, pp. 1135–1144. DOI: 10.1109/IV51971.2022.9827153 (cit. on pp. 48, 49).

[MN12] Philippe Morignot and Fawzi Nashashibi. "An ontology-based approach to relax traffic regulation for autonomous vehicle assistance". In: *CoRR* abs/1212.0768 (2012). DOI: 10.48550/arXiv.1212.0768 (cit. on pp. 48, 50).

[MP25] Kumar Manas and Adrian Paschke. "Knowledge Integration Strategies in Autonomous Vehicle Prediction and Planning: A Comprehensive Survey". In: (2025), pp. 694–701. DOI: 10.1109/IV64158.2025.11097371 (cit. on p. 148).

[Nal+20] Demin Nalic, Tomislav Mihalj, Maximilian Bäumler, Matthias Lehmann, Arno Eichberger, and Stefan Bernsteiner. "Scenario based testing of automated driving systems: A literature survey". In: *FISITA Web Congress, virtual, November 24, 2020*. Vol. 10. Bishops Stortford, UK: FISITA (UK) Limited, Nov. 2020, p. 1. URL: https://www.fisita.org/library/f2020-acm-096 (cit. on p. 18).

[NDW09] Kilian von Neumann-Cosel, Marius Dupuis, and Christian Weiss. "Virtual test drive – provision of a consistent tool-set for [D,H,S,V]-in-the-loop". In: *Proceedings of the Driving Simulation Conference Europe 2009, DSC2009, Monaco, February 4-6, 2009*. Bron, France: INRETS, 2009. URL: https://mediatum.ub.tum.de/1289102 (cit. on p. 35).

[Neu+20] Christian Neurohr, Lukas Westhofen, Tabea Henning, Thies de Graaff, Eike Möhlmann, and Eckard Böde. "Fundamental Considerations around Scenario-Based Testing for Automated Driving". In: *IEEE Intelligent Vehicles Symposium, IV 2020, Las Vegas, NV, USA, October 19 - November 13, 2020*. New York, NY, USA: IEEE, 2020, pp. 121–127. DOI: 10.1109/IV47402.2020.9304823 (cit. on pp. 5, 12, 17, 18, 34, 36–38, 163).

[Neu+21] Christian Neurohr, Lukas Westhofen, Martin Butz, Martin Herbert Bollmann, Ulrich Eberle, and Roland Galbas. "Criticality Analysis for the Verification and Validation of Automated Vehicles". In: *IEEE Access* 9 (2021), pp. 18016–18041. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3053159 (cit. on pp. 35, 136, 149).

[Neu+23] Birte Neurohr, Tjark Koopmann, Eike Möhlmann, and Martin Fränzle. "Determining the Validity of Simulation Models for the Verification of

Automated Driving Systems". In: *IEEE Access* 11 (2023), pp. 102949–102960. DOI: 10.1109/ACCESS.2023.3316354 (cit. on p. 38).

[Neu+26] Christian Neurohr, Lukas Westhofen, Tjark Koopmann, Eike Möhlmann, Eckard Böde, and Axel Hahn. *On Scenario Formalisms for Automated Driving*. Ed. by Andreas Rauh, Bernd Finkbeiner, and Paul Kröger. Berlin, Germany, 2026. DOI: 10.1007/978-3-032-16855-9_17 (cit. on pp. 25, 34, 36, 37).

[NR03] Jean-Marc Nigro and Michèle Rombaut. "IDRES: A rule-based system for driving situation recognition with uncertainty management". In: *Information Fusion* 4.4 (2003), pp. 309–317. DOI: 10.1016/S1566-2535(03)00042-3 (cit. on pp. 48, 50).

[Ogd68] William F. Ogden. "A Helpful Result for Proving Inherent Ambiguity". In: *Mathematical Systems Theory* 2.3 (1968), pp. 191–194. DOI: 10.1007/BF01694004 (cit. on p. 129).

[OK05] Dietmar Otte and Christian Krettek. "Rollover accidents of cars in the German road traffic–an In-depth-analysis of injury and deformation pattern by GIDAS". In: *Proceedings of the th International Technical Conference on the Enhanced Safety of Vehicles, ESV 2005, Washington, D.C., USA, June 6-9, 2005*. 05-0093. Washington, D.C., USA: NHTSA, 2005. URL: https://www-esv.nhtsa.dot.gov/Proceedings/19/05-0093-0.pdf (cit. on p. 31).

[ÖM14] Özgür Lütfü Özçep and Ralf Möller. "Ontology Based Data Access on Temporal and Streaming Data". In: *Reasoning on the Web in the Big Data Era - 10th International Summer School 2014, Athens, Greece, September 8-13, 2014*. Ed. by Manolis Koubarakis, Giorgos B. Stamou, Giorgos Stoilos, Ian Horrocks, Phokion G. Kolaitis, Georg Lausen, and Gerhard Weikum. Vol. 8714. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2014, pp. 279–312. DOI: 10.1007/978-3-319-10587-1_7 (cit. on p. 7).

[Ort10] Magdalena Ortiz. "Query answering in expressive description logics: Techniques and complexity results". PhD thesis. Wien, Austria: Technische Universität Wien, 2010. DOI: 20.500.12708/10546 (cit. on p. 108).

[Ozb+08] Kaan Ozbay, Hong Yang, Bekir Bartin, and Sandeep Mudigonda. "Derivation and validation of new simulation-based surrogate safety measure". In: *Transportation Research Record* 2083.1 (2008), pp. 105–113. DOI: 10.3141/2083-12 (cit. on p. 43).

[Pal+11] Rob Palin, David Ward, Ibrahim Habli, and Roger Rivett. "ISO 26262 safety cases: Compliance and assurance". In: *6th IET International Conference on System Safety, ICSS 2011, Birmingham, UK, September 20-22, 2011*. Hertfordshire, UK: IET, 2011, pp. 1–6. DOI: 10.1049/cp.2011.0251 (cit. on p. 18).

[Pan+25] Efimia Panagiotaki, Georgi Pramatarov, Lars Kunze, and Daniele De Martini. "GraphSCENE: On-Demand Critical Scenario Generation for Autonomous Vehicles in Simulation". In: *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hangzhou, China, October 19-25, 2025*. New York, NY, USA: IEEE, 2025, pp. 13801–13808. DOI: 10.1109/IROS60139.2025.11246040 (cit. on p. 148).

[PGD19] Thomas Ponn, Christian Gnandt, and Frank Diermeyer. "An optimization-based method to identify relevant scenarios for type approval of automated vehicles". In: *Proceedings of the 26th International Technical Conference on the Enhanced Safety of Vehicles, ESV 2019, Eindhoven, The Netherlands*. Washington, D.C., USA: NHTSA, June 2019, pp. 103–121. URL: https://www-esv.nhtsa.dot.gov/Proceedings/26/26ESV-000024.pdf (cit. on p. 37).

[PH68] Stuart R. Perkins and Joseph L. Harris. "Traffic conflict characteristics-accident potential at intersections". In: *Highway Research Record* (225 1968), pp. 35–43. URL: https://onlinepubs.trb.org/Onlinepubs/hrr/1968/225/225-004.pdf (cit. on p. 40).

[PHR13] Lakshmi N. Peesapati, Michael P. Hunter, and Michael O. Rodgers. "Evaluation of postencroachment time as surrogate for opposing left-turn crashes". In: *Transportation Research Record* 2386.1 (2013), pp. 42–51. DOI: 10.3141/2386-06 (cit. on p. 45).

[Pog+08] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. "Linking Data to Ontologies". In: *Journal on Data Semantics* 10 (2008). Ed. by Stefano Spaccapietra, pp. 133–173. DOI: 10.1007/978-3-540-77688-8_5 (cit. on p. 75).

[Pog+18] Fabian Poggenhans, Jan-Hendrik Pauls, Johannes Janosovits, Stefan Orf, Maximilian Naumann, Florian Kuhnt, and Matthias Mayr. "Lanelet2: A high-definition map framework for the future of automated driving". In: *21st IEEE International Conference on Intelligent Transportation Systems, ITSC 2018, Maui, HI, USA, November 4-7, 2018*. Ed. by Wei-Bin Zhang, Alexandre M. Bayen, Javier J. Sánchez Medina, and Matthew J. Barth. New York, NY, USA: IEEE, 2018, pp. 1672–1679. DOI: 10.1109/ITSC.2018.8569929 (cit. on p. 38).

[PPO22] Jason Priem, Heather A. Piwowar, and Richard Orr. "OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts". In: *CoRR* abs/2205.01833 (2022). DOI: 10.48550/arXiv.2205.01833 (cit. on p. 40).

[Put+23] Lina Putze, Lukas Westhofen, Tjark Koopmann, Eckard Böde, and Christian Neurohr. "On Quantification for SOTIF Validation of Automated Driving Systems". In: *IEEE Intelligent Vehicles Symposium, IV 2023, Anchorage, AK, USA, June 4-7, 2023*. New York, NY, USA: IEEE, 2023, pp. 1–8. DOI: 10.1109/IV55152.2023.10186795 (cit. on pp. 5, 11, 17, 18, 20, 22, 24, 26–29).

[PY82] Christos H. Papadimitriou and Mihalis Yannakakis. "The Complexity of Facets (and Some Facets of Complexity)". In: *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, San Francisco, California, USA, May 5-7, 1982*. Ed. by Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber. New York, NY, USA: ACM, 1982, pp. 255–260. DOI: 10.1145/800070.802199 (cit. on pp. 170, 171).

[RB06] Grigore Rosu and Saddek Bensalem. "Allen Linear (Interval) Temporal Logic - Translation to LTL and Monitor Synthesis". In: *Computer Aided Verification - 18th International Conference, CAV 2006, Seattle, WA, USA, August 17-20, 2006*. Ed. by Thomas Ball and Robert B. Jones. Vol. 4144. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2006, pp. 263–277. DOI: 10.1007/11817963_25 (cit. on p. 86).

[RFA18] Tscharn Robert, Naujoks Frederik, and Neukum Alexandra. "The perceived criticality of different time headways is depending on velocity". In: *Transportation Research Part F: Psychology and Behaviour* 58 (2018), pp. 1043–1052. DOI: 10.1016/j.trf.2018.08.001 (cit. on p. 33).

[RG10] Sebastian Rudolph and Birte Glimm. "Nominals, Inverses, Counting, and Conjunctive Queries or: Why Infinity is your Friend!" In: *Journal of Artificial Intelligence Research* 39 (2010), pp. 429–481. DOI: 10.1613/JAIR.3029 (cit. on pp. 77, 85).

[Rie+20] Stefan Riedmaier, Thomas Ponn, Dieter Ludwig, Bernhard Schick, and Frank Diermeyer. "Survey on Scenario-Based Safety Assessment of Automated Vehicles". In: *IEEE Access* 8 (2020), pp. 87456–87477. DOI: 10.1109/ACCESS.2020.2993730 (cit. on pp. 18, 163).

[Riz+17] Albert Rizaldi, Jonas Keinholz, Monika Huber, Jochen Feldle, Fabian Immler, Matthias Althoff, Eric Hilgendorf, and Tobias Nipkow. "Formalising and Monitoring Traffic Rules for Autonomous Vehicles in Isabelle/HOL". In: *Integrated Formal Methods - 13th International Conference, iFM 2017,*

Turin, Italy, September 20-22, 2017. Ed. by Nadia Polikarpova and Steve A. Schneider. Vol. 10510. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2017, pp. 50–66. DOI: 10.1007/978-3-319-66845-1_4 (cit. on pp. 48, 49).

[Rud11] Sebastian Rudolph. "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*. Ed. by Axel Polleres, Claudia d'Amato, Marcelo Arenas, Siegfried Handschuh, Paula Kroner, Sascha Ossowski, and Peter Patel-Schneider. Berlin, Germany: Springer, 2011, pp. 76–136. ISBN: 978-3-642-23032-5. DOI: 10.1007/978-3-642-23032-5_2 (cit. on p. 100).

[SAE21] SAE International. *J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Standard. Warrendale, USA, 2021. URL: https://www.sae.org/standards/j3016-taxonomy-definitions-terms-related-driving-automation-systems-road-motor-vehicles (cit. on p. 19).

[Sch+19] Karsten Scheibler, Andreas Eggers, Tino Teige, Marius Walz, Tom Bienmüller, and Udo Brockmeyer. "Solving Constraint Systems from Traffic Scenarios for the Validation of Autonomous Driving". In: *Proceedings of the 4th SC-Square Workshop co-located with the SIAM Conference on Applied Algebraic Geometry, SC-square@SIAM AG 2019, Bern, Switzerland, July 10, 2019*. Ed. by John Abbott and Alberto Griggio. Vol. 2460. CEUR Workshop Proceedings. CEUR-WS.org, 2019. URL: https://ceur-ws.org/Vol-2460/paper2.pdf (cit. on p. 37).

[Sch+21] Maike Scholtes, Lukas Westhofen, Lara Ruth Turner, Katrin Lotto, Michael Schuldes, Hendrik Weber, Nicolas Wagener, Christian Neurohr, Martin Bollmann, Franziska Körtke, Johannes Hiller, Michael Hoss, Julian Bock, and Lutz Eckstein. "6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment". In: *IEEE Access* 9 (2021), pp. 59131–59147. DOI: 10.1109/ACCESS.2021.3072739 (cit. on pp. 10, 12, 136, 148).

[Sch+22] Maike Scholtes, Michael Schuldes, Hendrik Weber, Nicolas Wagener, Michael Hoss, and Lutz Eckstein. "OMEGAFormat: A comprehensive format of traffic recordings for scenario extraction". In: *14. Workshop Fahrerassistenz und automatisiertes Fahren, Berkheim, Germany, May 9-11, 2022*. Karlsruhe, Germany: Uni-DAS e.V., 2022, pp. 195–205. ISBN: 978-3-941543-65-2. URL: https://www.uni-das.de/images/pdf/fas-workshop/2022/FAS2022-18-Scholtes.pdf (cit. on pp. 38, 52).

[Sch+23] Barbara Schütt, Joshua Ransiek, Thilo Braun, and Eric Sax. "1001 Ways of Scenario Generation for Testing of Self-driving Cars: A Survey". In: *IEEE Intelligent Vehicles Symposium, IV 2023, Anchorage, AK, USA, June 4-7, 2023*. New York, NY, USA: IEEE, 2023, pp. 1–8. DOI: 10.1109/IV55152.2023.10186735 (cit. on p. 148).

[Sch+24a] Till Schallau, Dominik Mäckel, Stefan Naujokat, and Falk Howar. "STARS: A Tool for Measuring Scenario Coverage When Testing Autonomous Robotic Systems". In: *Dependable Computing - EDCC 2024 Workshops - SafeAutonomy, TRUST in BLOCKCHAIN, Leuven, Belgium, April 8, 2024*. Ed. by Behrooz Sangchoolie, Rasmus Adler, Richard Hawkins, Philipp Schleiss, Alessia Arteconi, and Adriano Mancini. Vol. 2078. Communications in Computer and Information Science. Berlin, Germany: Springer, 2024, pp. 62–70. DOI: 10.1007/978-3-031-56776-6_6 (cit. on p. 136).

[Sch+24b] Till Schallau, Stefan Naujokat, Fiona Kullmann, and Falk Howar. "Tree-Based Scenario Classification - A Formal Framework for Measuring Domain Coverage When Testing Autonomous Systems". In: *NASA Formal Methods - 16th International Symposium, NFM 2024, Moffett Field, CA, USA, June 4-6, 2024*. Ed. by Nathaniel Benz, Divya Gopinath, and Nija Shi. Vol. 14627. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2024, pp. 259–278. DOI: 10.1007/978-3-031-60698-4_15 (cit. on pp. 48, 50, 148, 163, 165).

[Sch91] Klaus Schild. "A Correspondence Theory for Terminological Logics: Preliminary Report". In: *Proceedings of the 12th International Joint Conference on Artificial Intelligence, IJCAI 1991, Sydney, Australia, August 24-30, 1991*. Ed. by John Mylopoulos and Raymond Reiter. Burlington, MA, USA: Morgan Kaufmann, 1991, pp. 466–471. URL: http://ijcai.org/Proceedings/91-1/Papers/072.pdf (cit. on p. 66).

[SGE24] Michael Schuldes, Christoph Glasmacher, and Lutz Eckstein. "scenario.center: Methods from Real-world Data to a Scenario Database". In: *IEEE Intelligent Vehicles Symposium, IV 2024, Jeju Island, Republic of Korea, June 2-5, 2024*. New York, NY, USA: IEEE, 2024, pp. 1119–1126. DOI: 10.1109/IV55156.2024.10588866 (cit. on p. 52).

[She11] Steven G Shelby. "Delta-V as a measure of traffic conflict severity". In: *90th Annual Meeting of the Transportation Research Board, Washington, D.C., USA, January 23-27, 2011*. Washington, D.C., USA: Transportation Research Board, 2011, pp. 14–16. URL: https://trid.trb.org/View/1093469 (cit. on p. 46).

[Sir+07] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. "Pellet: A practical OWL-DL reasoner". In: *Journal of Web Semantics* 5.2 (2007), pp. 51–53. ISSN: 1570-8268. DOI: 10.1016/j.websem.2007.03.004 (cit. on pp. 84, 137).

[Sir06] Evren Sirin. "Combining Description Logic Reasoning with AI Planning for Composition of Web Services". PhD thesis. College Park, MD, USA: University of Maryland, 2006. ISBN: 0-542-96121-0. URL: https://hdl.handle.net/1903/4070 (cit. on pp. 78, 82, 83, 122, 143).

[SM23] Megan Strauss and Stefan Mitsch. "Slow Down, Move Over: A Case Study in Formal Verification, Refinement, and Testing of the Responsibility-Sensitive Safety Model for Self-Driving Cars". In: *Tests and Proofs - 17th International Conference, TAP 2023, Leicester, UK, July 18-19, 2023*. Ed. by Virgile Prevosto and Cristina Seceleanu. Vol. 14066. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2023, pp. 149–167. DOI: 10.1007/978-3-031-38828-6_9 (cit. on p. 44).

[SP04] Evren Sirin and Bijan Parsia. "Pellet: An OWL DL Reasoner". In: *Proceedings of the 2004 International Workshop on Description Logics, DL 2004, Whistler, British Columbia, Canada, June 6-8, 2004*. Ed. by Volker Haarslev and Ralf Möller. Vol. 104. CEUR Workshop Proceedings. Aachen, Germany: CEUR-WS.org, 2004. URL: https://ceur-ws.org/Vol-104/30Sirin-Parsia.pdf (cit. on p. 137).

[SP06] Evren Sirin and Bijan Parsia. "Optimizations for Answering Conjunctive ABox Queries". In: *Proceedings of the 2006 International Workshop on Description Logics, DL 2006, Windermere, Lake District, UK, May 30-June 1, 2006*. Ed. by Bijan Parsia, Ulrike Sattler, and David Toman. Vol. 189. CEUR Workshop Proceedings. Aachen, Germany: CEUR-WS.org, 2006. URL: https://ceur-ws.org/Vol-189/submission_32.pdf (cit. on pp. 81, 82, 108).

[SSS17] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. "On a Formal Model of Safe and Scalable Self-driving Cars". In: *CoRR* abs/1708.06374 (2017). DOI: 10.48550/arXiv.1708.06374 (cit. on p. 44).

[Ste+25] Ralf Stemmer, Ishan Saxena, Lukas Panneke, Dominik Grundt, Anna Austel, Eike Möhlmann, and Bernd Westphal. "Runtime monitoring of complex scenario-based requirements for autonomous driving functions". In: *Science of Computer Programming* 244 (2025), p. 103301. DOI: 10.1016/J.SCICO.2025.103301 (cit. on pp. 48, 50).

[Ste21] Stephan Immen. *KBA erteilt erste Genehmigung zum automatisierten Fahren*. Press release no. 49/2021. Accessed on 1st of September 2025. Flensburg, Germany, Dec. 2021. URL: https://www.kba.de/DE/Presse/Pressemitteilungen/Allgemein/2021/pm49_2021_erste_Genehmigung_automatisiertes_Fahren.html (cit. on p. 19).

[Ste24] Bernhard Steffen, ed. *Bridging the Gap Between AI and Reality - First International Conference, AISoLA 2023, Crete, Greece, October 23-28, 2023, Proceedings*. Vol. 14380. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2024. ISBN: 978-3-031-46001-2. DOI: 10.1007/978-3-031-46002-9 (cit. on p. 52).

[Str06] Arbeitsgruppe Straßenentwurf. *Richtlinien für die Anlage von Stadtstrassen (RASt 06)*. Forschungsgesellschaft für das Strassenwesen (Germany), 2006. ISBN: 978-3-939715-21-4 (cit. on p. 3).

[Stu15] Studiengesellschaft für den Kombinierten Verkehr e.V. (SGKV). *Rechtliche Vorschriften zu Eis und Schnee auf Ladeeinheiten*. Technical Report. Berlin, Germany: SGKV – Studiengesellschaft für den Kombinierten Verkehr e.V., Mar. 2015. URL: https://sgkv.de/wp-content/uploads/2020/07/Survey-Recht-bei-Eis-und-Schnee-auf-LE.pdf (cit. on p. 30).

[Sun+22] Jian Sun, He Zhang, Huajun Zhou, Rongjie Yu, and Ye Tian. "Scenario-Based Test Automation for Highly Automated Vehicles: A Review and Paving the Way for Systematic Safety Assurance". In: *IEEE Transactions on Intelligent Transportation Systems* 23.9 (2022), pp. 14088–14103. DOI: 10.1109/TITS.2021.3136353 (cit. on p. 163).

[Sve98] Åse Svensson. "A method for analysing the traffic process in a safety perspective". PhD thesis. Lund, Sweden: Lund Institute of Technology, 1998. URL: https://portal.research.lu.se/en/publications/fe7d733d-4108-48ea-bebe-86fef046ad51 (cit. on p. 40).

[SW25] Dominik Schmid and Lukas Westhofen. *Data from: Test Coverage of Automated Robotic Systems in Open World Environments*. Dataset. Zenodo, Dec. 2025. DOI: 10.5281/zenodo.17775280 (cit. on p. 175).

[SWH26] Ingo Stierand, Lukas Westhofen, and Willem Hagemann. "On Using Ontologies in the Engineering of Intelligent Cyber-Physical Systems". In: *Engineering Safe and Trustworthy Cyber Physical Systems: Essays Dedicated to Werner Damm on the Occasion of His 71st Birthday*. Ed. by Martin Fränzle, Jürgen Niehaus, and Bernd Westphal. Berlin, Germany: Springer, 2026, pp. 54–75. ISBN: 978-3-031-97537-0. DOI: 10.1007/978-3-031-97537-0_4 (cit. on p. 19).

[TDB11] Andreas Tamke, Thao Dang, and Gabi Breuel. "A flexible method for criticality assessment in driver assistance systems". In: *IEEE Intelligent Vehicles Symposium, IV 2011, Baden-Baden, Germany, June 5-9, 2011*. New York, NY, USA: IEEE, 2011, pp. 697–702. DOI: 10.1109/IVS.2011.5940482 (cit. on p. 46).

[THÖ15] Veronika Thost, Jan Holste, and Özgür L. Özçep. "On Implementing Temporal Query Answering in DL-Lite (extended abstract)". In: *Proceedings of the 28th International Workshop on Description Logics, DL 2015, Athens, Greece, June 7-10, 2015*. Ed. by Diego Calvanese and Boris Konev. Vol. 1350. CEUR Workshop Proceedings. Aachen, Germany: CEUR-WS.org, 2015. URL: https://ceur-ws.org/Vol-1350/paper-63.pdf (cit. on p. 86).

[Tob00] Stephan Tobies. "The Complexity of Reasoning with Cardinality Restrictions and Nominals in Expressive Description Logics". In: *Journal of Artificial Intelligence Research* 12 (2000), pp. 199–217. DOI: 10.1613/JAIR.705 (cit. on pp. 72, 80).

[Tol+24] Felipe Toledo, Trey Woodlief, Sebastian G. Elbaum, and Matthew B. Dwyer. "Specifying and Monitoring Safe Driving Properties with Scene Graphs". In: *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*. New York, NY, USA: IEEE, 2024, pp. 15577–15584. DOI: 10.1109/ICRA57147.2024.10610973 (cit. on pp. 48, 50).

[TSW15] Ahmed Tageldin, Tarek Sayed, and Xuesong Wang. "Can time proximity measures be used as safety indicators in all driving cultures? Case study of motorcycle safety in China". In: *Transportation Research Record* 2520.1 (2015), pp. 165–174. DOI: 10.3141/2520-19 (cit. on pp. 43, 45).

[Ulb+15] Simon Ulbrich, Till Menzel, Andreas Reschka, Fabian Schuldt, and Markus Maurer. "Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving". In: *18th IEEE International Conference on Intelligent Transportation Systems, ITSC 2015, Gran Canaria, Spain, September 15-18, 2015*. New York, NY, USA: IEEE, 2015, pp. 982–988. DOI: 10.1109/ITSC.2015.164 (cit. on pp. 23–25, 34).

[Uni21] United Nations Economic Commission for Europe (UNECE). *UN Regulation No. 157: Uniform provisions concerning the approval of vehicles with regard to Automated Lane Keeping Systems*. UN Regulation. Geneva, Switzerland, 2021. URL: https://unece.org/sites/default/files/2023-12/R157e.pdf (cit. on pp. 2, 19, 20, 166).

[Van90] Adrianus Rigardus Antonius Van der Horst. "A time-based analysis of road user behaviour in normal and critical encounters". PhD thesis. Delft, Netherlands: TU Delft, 1990. ISBN: 90-90-03340-8. URL: https://resolver.tudelft.nl/uuid:8fb40be7-fae1-4481-bc37-12a7411b85c7 (cit. on p. 40).

[Var82] Moshe Y. Vardi. "The Complexity of Relational Query Languages (Extended Abstract)". In: *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, San Francisco, California, USA, May 5-7, 1982*. Ed. by Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber. New York, NY, USA: ACM, 1982, pp. 137–146. DOI: 10.1145/800070.802186 (cit. on pp. 61, 166).

[Ver13] Bundesministerium für Verkehr. *Straßenverkehrs-Ordnung (StVO)*. German. Legal Act. Köln, Germany, 2013. URL: https://www.gesetze-im-internet.de/stvo_2013/ (cit. on p. 33).

[Ver22a] Bundesministerium für Verkehr. *Verordnung zur Genehmigung und zum Betrieb von Kraftfahrzeugen mit autonomer Fahrfunktion in festgelegten Betriebsbereichen (AFGBV)*. German. Legal Act. Köln, Germany, 2022. URL: https://www.gesetze-im-internet.de/afgbv/ (cit. on pp. 19, 20).

[Ver22b] Bundesministerium für Verkehr. *Verordnung zur Genehmigung und zum Betrieb von Kraftfahrzeugen mit autonomer Fahrfunktion in festgelegten Betriebsbereichen (AFGBV) Anlage 1 Anforderungen an Kraftfahrzeuge mit autonomer Fahrfunktion*. German. Legal Act. Köln, Germany, 2022. URL: https://www.gesetze-im-internet.de/afgbv/anlage_1.html (cit. on pp. 163, 178).

[Vog03] Katja Vogel. "A comparison of headway and time to collision as safety indicators". In: *Accident Analysis & Prevention* 35.3 (2003), pp. 427–433. DOI: 10.1016/S0001-4575(02)00022-2 (cit. on p. 45).

[Wał+20] Przemysław Andrzej Wałęga, Bernardo Cuenca Grau, Mark Kaminski, and Egor V. Kostylev. "DatalogMTL over the Integer Timeline". In: *Principles of Knowledge Representation and Reasoning: Proceedings of the 17th International Conference, KR 2020, Rhodes, Greece, September 12-18, 2020*. Ed. by Diego Calvanese, Esra Erdem, and Michael Thielscher. Palo Alto, USA: IJCAI/AAAI Press, 2020, pp. 768–777. ISBN: 978-0-9992411-7-2. DOI: 10.24963/kr.2020/79 (cit. on p. 86).

[Wan+22] Dingmin Wang, Pan Hu, Przemysław Andrzej Wałęga, and Bernardo Cuenca Grau. "MeTeoR: Practical Reasoning in Datalog with Metric Temporal Operators". In: *The Thirty-Sixth AAAI Conference on Artificial Intelligence,*

AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022, Virtual Event, February 22-March 1, 2022. Palo Alto, CA, USA: AAAI Press, 2022, pp. 5906–5913. DOI: 10.1609/AAAI.V36I5.20535 (cit. on pp. 3, 86, 145).

[Wan+24a] Cheng Wang, Kai Storms, Ning Zhang, and Hermann Winner. "Runtime unknown unsafe scenarios identification for SOTIF of autonomous vehicles". In: *Accident Analysis & Prevention* 195 (2024), p. 107410. DOI: 10.1016/j.aap.2023.107410 (cit. on p. 148).

[Wan+24b] Hong Wang, Wenbo Shao, Chen Sun, Kai Yang, Dongpu Cao, and Jun Li. "A survey on an emerging safety challenge for autonomous vehicles: Safety of the intended functionality". In: *Engineering* 33 (2024), pp. 17–34. DOI: 10.1016/j.eng.2023.10.011 (cit. on p. 18).

[WB25] Jasmine Wu and Deirdre Bosa. *Waymo crosses 450,000 weekly paid rides as Alphabet robotaxi unit widens lead on Tesla*. Accessed on 4th of January 2026. Dec. 2025. URL: https://www.cnbc.com/amp/2025/12/08/waymo-paid-rides-robotaxi-tesla.html (cit. on p. 2).

[WBK20] Lukas Westhofen, Philipp Berger, and Joost-Pieter Katoen. "Benchmarking Software Model Checkers on Automotive Code". In: *NASA Formal Methods - 12th International Symposium, NFM 2020, Moffett Field, CA, USA, May 11-15, 2020*. Ed. by Ritchie Lee, Susmit Jha, and Anastasia Mavridou. Vol. 12229. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2020, pp. 133–150. DOI: 10.1007/978-3-030-55754-6_8 (cit. on p. 19).

[Wes+22a] Lukas Westhofen, Christian Neurohr, Martin Butz, Maike Scholtes, and Michael Schuldes. "Using ontologies for the formalization and recognition of criticality for automated driving". In: *IEEE Open Journal of Intelligent Transportation Systems* 3 (2022), pp. 519–538. DOI: 10.1109/OJITS.2022.3187247 (cit. on pp. 10, 11, 52, 53, 136, 148, 149).

[Wes+22b] Lukas Westhofen, Ingo Stierand, Jan Steffen Becker, Eike Möhlmann, and Willem Hagemann. "Towards a Congruent Interpretation of Traffic Rules for Automated Driving – Experiences and Challenges". In: *Proceedings of the International Workshop on Methodologies for Translating Legal Norms into Formal Representations (LN2FR 2022) in association with the 35th International Conference on Legal Knowledge and Information Systems (JURIX 2022), Saarbrücken, Germany, 2022, December 15-16, 2022*. Ithaca, NY, USA: arXiv, 2022, pp. 8–21. URL: https://arxiv.org/pdf/2305.12203#page=13 (cit. on pp. 3, 19).

[Wes+23] Lukas Westhofen, Christian Neurohr, Tjark Koopmann, Martin Butz, Barbara Schütt, Fabian Utesch, Birte Neurohr, Christian Gutenkunst, and Eckard Böde. "Criticality Metrics for Automated Driving: A Review and Suitability Analysis of the State of the Art". In: *Archives of Computational Methods in Engineering* 30.1 (Jan. 2023), pp. 1–35. ISSN: 1886-1784. DOI: 10.1007/s11831-022-09788-7 (cit. on pp. 5, 11, 17, 18, 33, 39–43, 47).

[Wes+24a] Lukas Westhofen, Christian Neurohr, Jean Christoph Jung, and Daniel Neider. "Answering Temporal Conjunctive Queries over Description Logic Ontologies for Situation Recognition in Complex Operational Domains". In: *Tools and Algorithms for the Construction and Analysis of Systems - 30th International Conference, TACAS 2024, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2024, Luxembourg City, Luxembourg, April 6-11, 2024, Proceedings Part I*. Ed. by Bernd Finkbeiner and Laura Kovács. Vol. 14570. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2024, pp. 167–187. DOI: 10.1007/978-3-031-57246-3_10 (cit. on pp. 8, 10, 11, 52, 58, 84, 89–91, 93, 96, 99–101, 104, 105, 108, 136, 147).

[Wes+24b] Lukas Westhofen, Christian Neurohr, Jean Christoph Jung, and Daniel Neider. "Topllet: An Optimized Engine for Answering Metric Temporal Conjunctive Queries". In: *NASA Formal Methods - 16th International Symposium, NFM 2024, Moffett Field, CA, USA, June 4-6, 2024*. Ed. by Nathaniel Benz, Divya Gopinath, and Nija Shi. Vol. 14627. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2024, pp. 314–321. DOI: 10.1007/978-3-031-60698-4_18 (cit. on pp. 9, 11, 84, 136, 145).

[Wes+26] Lukas Westhofen, Till Schallau, Dominik Schmid, Stefan Naujokat, Falk Howar, and Daniel Neider. "Test Coverage of Automated Robotic Systems in Open World Environments". In: *Formal Methods - 27th International Symposium, FM 2026, Tokyo, Japan, May 18-22, 2026, Proceedings, Part I*. Ed. by Augusto Sampaio and Marielle Stoelinga. Vol. 16556. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2026, pp. 130–150. DOI: 10.1007/978-3-032-26204-2_7 (cit. on pp. 11, 136, 164, 170, 173, 174, 179).

[Wes23a] Lukas Westhofen. *Openllet Temporal Query Benchmarks*. Version v0.0.1. Dec. 2023. DOI: 10.5281/zenodo.10436065 (cit. on p. 155).

[Wes23b] Lukas Westhofen. *Situation Recognition in Complex Operational Domains using Temporal and Description Logics – A Motivation from the Automotive Domain*. Version v1. Oct. 2023. DOI: 10.5281/zenodo.13859450 (cit. on p. 52).

[Wes24] Lukas Westhofen. *Experiments for 'Temporal Conjunctive Query Answering via Rewriting'*. Version v1.0.0. Dec. 2024. DOI: 10.5281/zenodo.14412131 (cit. on p. 155).

[Wes25a] Lukas Westhofen. *MLTL2LTLf v0.3*. Dec. 2025. DOI: 10.5281/zenodo.18097070 (cit. on p. 140).

[Wes25b] Lukas Westhofen. *STARS Logic MTCQ v2*. Dec. 2025. DOI: 10.5281/zenodo.17867411 (cit. on pp. 179, 180).

[Wes26a] Lukas Westhofen. *owlready2-augmentator v0.1*. Jan. 2026. DOI: 10.5281/zenodo.18214556 (cit. on p. 150).

[Wes26b] Lukas Westhofen. *pyauto v1.0.2*. Jan. 2026. DOI: 10.5281/zenodo.18214532 (cit. on p. 150).

[Wes26c] Lukas Westhofen. *Software for MTCQ evaluation on the inD data set*. Jan. 2026. DOI: 10.5281/zenodo.18097120 (cit. on p. 181).

[Wes26d] Lukas Westhofen. *The Traffic Ontology Benchmark*. Jan. 2026. DOI: 10.5281/zenodo.18214594 (cit. on p. 148).

[Wes26e] Lukas Westhofen. *Topllet v1.0.1*. Jan. 2026. DOI: 10.5281/zenodo.18096984 (cit. on p. 137).

[Wes26f] Lukas Westhofen. *topllet-benchmarks*. Jan. 2026. DOI: 10.5281/zenodo.18214573 (cit. on p. 145).

[WFE20] Nico Weber, Dirk Frerichs, and Ulrich Eberle. "A simulation-based, statistical approach for the derivation of concrete scenarios for the release of highly automated driving functions". In: *AmE 2020 – Automotive Meets Electronics 11th GMM-Symposium*. Berlin, Germany: VDE Verlag GmbH, 2020, pp. 1–6. ISBN: 3-8007-5202-6. URL: https://ieeexplore.ieee.org/abstract/document/9094569 (cit. on p. 37).

[WJN25] Lukas Westhofen, Jean Christoph Jung, and Daniel Neider. "Temporal Conjunctive Query Answering via Rewriting". In: *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*. Ed. by Toby Walsh, Julie Shah, and Zico Kolter. Palo Alto, CA, USA: AAAI Press, 2025, pp. 15221–15229. DOI: 10.1609/AAAI.V39I14.33670 (cit. on pp. 11, 84, 96, 114, 115, 117–120, 124–128, 136).

[Wor12] World Wide Web Consortium. *OWL 2 Web Ontology Language, Structural Specification and Functional-Style Syntax (Second Edition)*. Ed. by Boris Motik, Peter F. Patel-Schneider, and Bijan Parsia. Accessed on 4th of November 2024. Dec. 2012. URL: https://www.w3.org/TR/owl2-syntax/ (cit. on p. 6).

[WS14] Chen Wang and Nikiforos Stamatiadis. "Evaluation of a simulation-based surrogate safety metric". In: *Accident Analysis & Prevention* 71 (2014), pp. 82–92. DOI: 10.1016/j.aap.2014.05.004 (cit. on p. 43).

[WW16] Walther Wachenfeld and Hermann Winner. "The Release of Autonomous Vehicles". In: *Autonomous Driving: Technical, Legal and Social Aspects*. Ed. by Markus Maurer, J. Christian Gerdes, Barbara Lenz, and Hermann Winner. Berlin, Germany: Springer, 2016, pp. 425–449. ISBN: 978-3-662-48847-8. DOI: 10.1007/978-3-662-48847-8_21 (cit. on pp. 2, 18, 29, 34).

[Xia22] Mingyu Xiao. "An Exact MaxSAT Algorithm: Further Observations and Further Improvements". In: *Proceedings of the 31st International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July, 2022*. Ed. by Luc De Raedt. Palo Alto, CA, USA: IJCAI/AAAI Press, 2022, pp. 1887–1893. DOI: 10.24963/IJCAI.2022/262 (cit. on p. 172).

[Xu+21] Xiaoyan Xu, Xuesong Wang, Xiangbin Wu, Omar Hassanin, and Chen Chai. "Calibration and evaluation of the Responsibility-Sensitive Safety model of autonomous car-following maneuvers using naturalistic driving study data". In: *Transportation Research Part C: Emerging Technologies* 123 (2021), p. 102988. DOI: 10.1016/j.trc.2021.102988 (cit. on p. 44).

[Zha+16] Lihua Zhao, Ryutaro Ichise, Yutaka Sasaki, Zheng Liu, and Tatsuya Yoshikawa. "Fast decision making using ontology-based knowledge base". In: *2016 IEEE Intelligent Vehicles Symposium, IV 2016, Gotenburg, Sweden, June 19-22, 2016*. New York, NY, USA: IEEE, 2016, pp. 173–178. DOI: 10.1109/IVS.2016.7535382 (cit. on pp. 48, 51).

[Zha+24] Ce Zhang, Liang Hong, Dan Wang, Xinchao Liu, Jinzhe Yang, and Yier Lin. "Research on Driving Scenario Knowledge Graphs". In: *Applied Sciences* 14.9 (2024), p. 3804. DOI: 10.3390/app14093804 (cit. on p. 148).

[ZHM97] Hong Zhu, Patrick A. V. Hall, and John H. R. May. "Software Unit Test Coverage and Adequacy". In: *ACM Computing Surveys* 29.4 (1997), pp. 366–427. DOI: 10.1145/267580.267590 (cit. on p. 163).

[Zhu+17] Shufang Zhu, Lucas M. Tabajara, Jianwen Li, Geguang Pu, and Moshe Y. Vardi. "Symbolic LTLf Synthesis". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. Ed. by Carles Sierra. ijcai.org, 2017, pp. 1362–1369. DOI: 10.24963/IJCAI.2017/189 (cit. on p. 103).

[Zhu+22] Zhijing Zhu, Robin Philipp, Constanze Hungar, and Falk Howar. "Systematization and Identification of Triggering Conditions: A Preliminary Step for Efficient Testing of Autonomous Vehicles". In: *2022 IEEE Intelligent Vehicles Symposium, IV 2022, Aachen, Germany, June 4-9, 2022*. New York, NY, USA: IEEE, 2022, pp. 798–805. DOI: 10.1109/IV51971.2022.9827238 (cit. on p. 20).

[ZKZ23] Maximilian Zipfl, Nina Koch, and J. Marius Zöllner. "A Comprehensive Review on Ontologies for Scenario-based Testing in the Context of Autonomous Driving". In: *IEEE Intelligent Vehicles Symposium, IV 2023, Anchorage, AK, USA, June 4-7, 2023*. New York, NY, USA: IEEE, 2023, pp. 1–7. DOI: 10.1109/IV55152.2023.10186681 (cit. on pp. 53, 136, 148).