Maschinelles Lernen

LS–8 Report 1

Katharina Morik,[1]
Universität Dortmund,
Fachbereich Informatik,
Lehrstuhl VIII
e-mail: morik@kimo.informatik.uni-dortmund.de

Dortmund, 22. Juni 1993

[1]Dieser Text entspricht dem gleichnamigen Kapitel in dem von Günther Görz herausgegebenen Buch *Künstliche Intelligenz*, das bei Addison-Wesley erscheint.

Abstract

This report gives an overview of machine learning. The report concentrates on methods rather than on the large number of systems. The logic-based approaches are described in some detail. The main paradigms are indicated and used for presenting practical techniques in a unified way.

Inhaltsverzeichnis

1 Einleitung

Das maschinelle Lernen gehört zu den Fähigkeiten, deren Verfügbarkeit auf einem Rechner bereits als Ziel formuliert wurde, als der erste praktische Rechnereinsatz mit ENIAC in Philadelphia gelungen war. Die Idee dabei war, daß Programmierer von Routinearbeiten entlastet und Programme schneller erstellt werden sollten. Für Alan Turing war die Lernfähigkeit eines Rechners die wichtigste Intelligenzleistung [Turing, 1987]. Er empfahl, einen Rechner "zu erziehen", so daß er seine Leistungen verbessert, da man unmöglich alles einprogrammieren könne. Insofern waren maschinelles Lernen und Programmsynthese damals noch nicht unterschieden. Diese Unterscheidung kam erst mit den Produktionensystemen, die ja "Wissen" von dem Verarbeitungsprogramm (Interpreter) unterschieden. Maschinelles Lernen wurde dann eingegrenzt auf den automatischen Erwerb von Regeln oder die Verbesserung von Regelmengen. Diese Eingrenzung ergab sich zum einen aus der psychologischen Motivation von Produktionensystemen. Produktionen wurden ja von [Simon, 1978] gerade deshalb als adäquate Repräsentation kognitiver Einheiten ausgewählt, weil sie dadurch, daß sie klein, gleich strukturiert und nicht miteinander verzahnt sind, leichter erlernbar seien. Zum anderen folgte die Eingrenzung aus der praktischen Notwendigkeit, den Regelerwerb für Expertensysteme zu unterstützen. Heute sind Produkte zum induktiven Regelerwerb auf dem Markt erhältlich und das Gebiet ist wieder breiter, in seinen Fragestellungen vielfältiger geworden. So wird an der theoretischen Frage gearbeitet, unter welchen Bedingungen Lernen mit welchem Aufwand möglich ist. Oder es wird der Zusammenhang zwischen Lernen und nicht-monotonem Schließen oder der zwischen Lernen und Wissensrevision herausgearbeitet, um nur einige Beispiele herauszugreifen. Oder es werden ganz praktisch Lernverfahren in so verschiedene Gebiete wie terminologische Logiken (KL-ONE), Datenbanken oder Robotik integriert.

Auf der *International Joint Conference on Artificial Intelligence* 1991 in Sidney war das Gebiet des maschinellen Lernens mit 37 Vorträgen vertreten,[1] die zu diesem Thema angekündigt waren. Unter Vorträgen, die anderen Gebieten zugeordnet waren, gab es zusätzlich solche, die sich mit maschinellem Lernen beschäftigten. Bedenkt man, daß es außerdem regelmäßige Konferenzen zum Thema gibt, die jeweils von etwa 200 Teilnehmern besucht werden, sowie unregelmäßige, spezialisierte Arbeitstreffen mit etwa 30-60 Teilnehmern, erkennt man den Stellenwert dieses Gebietes. Die wichtigen regelmäßigen Konferenzen sind:

- International Conference/Workshop on Machine Learning (IML), eine jährlich stattfindende internationale Veranstaltung, die abwechselnd als Reihe von Arbeitstreffen mit einigen Plenarvorträgen und als Konferenz durchgeführt wird;

- European Conference on Machine Learning (ECML), eine eineinhalbjährlich durchgeführte europäische Tagung;

[1] An zweiter Stelle folgt Wissensrepräsentation mit 36 Beiträgen.

- Computational Learning Theory (CoLT), eine jährliche, überwiegend amerikanische Konferenz zur Theorie des maschinellen Lernens;
- Algorithmic Learning Theory (ALT), eine jährliche, japanisch-europäische Konferenz zur Theorie maschinellen Lernens.

Hinzu kommen Sommerschulen, die eine Einführung in das Gebiet geben. Eine Zeitschrift, die mit 12 Ausgaben im Jahr die wichtigsten Ergebnisse präsentiert (*Machine Learning Journal*) rundet die Infrastruktur des Gebietes ab.

2 Was ist Lernen?

Die erste Frage, die meist gestellt wird, ist, wie wir Lernen definieren können, so daß wir einem Rechner eine – eingeschränkte – Lernfähigkeit zusprechen können. Die bekannte Definition von [Simon, 1983] lautet:

> *Lernen ist jede Veränderung eines Systems, die es ihm erlaubt, eine Aufgabe bei der Wiederholung derselben Aufgabe oder einer Aufgabe derselben Art besser zu lösen.*

Diese Definition ist aus zwei Gründen kritisiert worden: sie deckt auch solche Phänomene ab, die man üblicherweise nicht als Lernen bezeichnet, und sie deckt nicht alle dem Lernen zugerechneten Phänomene ab. Ein Beispiel dafür, daß Lernen nicht der einzige Grund für eine verbesserte Leistung ist, stammt von [Michalski, 1986]. Wenn es die Aufgabe ist, etwas zu schneiden, so wird die Leistung dadurch verbessert, daß man ein schärferes Messer nimmt. Das Messerschärfen ist aber kein Lernen. Nur das Herausfinden, daß mithilfe eines schärferen Messers das Schneiden zu verbessern ist, wäre Lernen. Simons Definition kann diese beiden Fälle aber nicht unterscheiden. Auch das zufällige Verwenden eines schärferen Messers würde seine Definition erfüllen. Die Lernfähigkeit von Programmen könnte gemäß Simons Definition dadurch nachgewiesen werden, daß wir dasselbe Programm auf einem schnelleren Rechner laufen ließen. Das System, Rechner und Programm, würde dann dieselbe Aufgabe schneller lösen. Als Verbesserung der Definition könnte man vorschlagen, daß alle Teile eines Systems verändert werden, um die Leistung zu steigern. Das würde dann aber das Hinzufügen einer Regel und die dadurch gesteigerte Leistung eines regelbasierten Systems bei gleichbleibendem Interpreter ausschließen. Man hätte dann gerade die Methode ausgeschlossen, die maschinelles Lernen erst ermöglichte, nämlich die Trennung von lernbaren Einheiten und nicht-lernbarer Verarbeitung.

Michalski gibt auch ein drastisches Beispiel dafür an, daß Leistungssenkung ein Lernergebnis sein kann [Michalski, 1986]. Wenn Zwangsarbeiter einen Weg finden, wie sie weniger leisten können, so wäre das ein Beispiel für Lernen. Sie könnten lernen, wie man weniger tut und doch gleich beschäftigt aussieht. Damit weist Michalski auf die Zielabhängigkeit des Leistungsbegriffs hin. Die Arbeiter verbessern ihre Leistung der Vortäuschung und verschlechtern ihre Arbeitsleistung. Je nachdem, wie man die Aufgabe definiert, fällt ihre Tätigkeit unter Simons Definition, oder nicht. Scott argumentiert gegen die Leistungsmessung bei der Definition vom Lernen [Scott, 1983]. Er führt als Beispiel einen Spaziergänger in einer ihm noch unbekannten Stadt an, der an der öffentlichen Bibliothek vorbeikommt. Während er diese wahrnimmt, lernt er etwas über die Stadt, ohne irgendeine Aufgabe zu haben, für deren Lösung er wissen muß, ob und wo es eine Bibliothek gibt. Erst wenn ein Passant ihn nach dem Weg zur Bibliothek fragt, kann er das Gelernte einsetzen – und zwar schon beim ersten Passanten, nicht erst bei der Wiederholung. Simon hat einen Test angegeben, der auch bei dem Spaziergänger ergeben würde, daß der gelernt hätte, jedoch keine Definition. Der Test gehört nicht zum Lernen selbst. Der Spaziergänger lernt unabhängig davon, ob er getestet wird. [Scott, 1983] definiert Lernen ohne Rückgriff auf eine gegebene Leistung:

> *Lernen ist ein Prozeß, bei dem ein System eine abrufbare Repräsentation von vergangenen Interaktionen mit seiner Umwelt aufbaut.*

Damit ist die Leistung potentiell beobachtbar, weil die neue Repräsentation abrufbar ist. Das Lernen selbst ist aber unabhängig davon, ob sein Ergebnis jemals gebraucht wird. Auch wird eine Leistungssenkung durch Lernen nicht ausgeschlossen. So könnte jemand, der nur eine einzige Aussage über etwas weiß, wenn genau nach dieser gefragt wird, womöglich schneller antworten als jemand, der erst aus der Fülle seiner Informationen die passende heraussuchen muß. Ähnlich ist auch Michalskis Definition [Michalski, 1986]:

> *Lernen ist das Konstruieren oder Verändern von Repräsentationen von Erfahrungen.*

Beide Definitionen setzen einen Prozeß voraus, der Repräsentationen verwendet. Wieweit dieser durch Lernen aufgebaut oder verändert wird, bleibt offen. Schon aus dieser kurzen Diskussion über die Definition von Lernen wird deutlich, daß Lernen ähnlich schwierig zu fassen ist wie Intelligenz. Es bleibt unser umgangssprachliches Verständnis von dem, was für uns Lernen ist, als Anregung und als Richtschnur.

3 Drei Motivationen für das maschinelle Lernen

Maschinelles Lernen hat – wie alle anderen Teilgebiete der KI – drei verschiedene Motivationen: eine kognitionswissenschaftliche, eine theoretisch-technische und eine praktische, anwendungsorientierte. Für das maschinelle Lernen sind die einzelnen Ziele:

- Prinzipien menschlichen Lernens sollen mithilfe von operationalen Modellen untersucht werden.

- Insbesondere der induktive Schluß soll operationalisiert werden, aber auch die Verwendung anderer Schlußfolgerungen (Deduktion und Abduktion) zum Lernen soll untersucht werden.

- Die Arbeit am Rechner soll durch dessen Lernfähigkeit dem Benutzer erleichtert werden.

Nach einem kurzen Überblick über diese drei Ausrichtungen konzentrieren sich die darauf folgenden Abschnitte auf Verfahren, also den zweiten Aspekt.

3.1 Menschliches und maschinelles Lernen

Die kognitive Orientierung verwendet psychologische Arbeiten zur Begriffsbildung. Die Struktur, Verwendung und der Erwerb von Begriffen bei Kindern sind Gegenstand vieler Untersuchungen. Hier werden nur einige zusammengefaßt, um einen Einblick in wichtige Fragestellungen zu geben. Literatur zum Einstieg in dieses Thema wird am Ende des Abschnitts angeführt.

Die Begriffsbildung kann in zwei Phänomenbereiche unterteilt werden: die Aggregation und die Charakterisierung oder Definition. Die **Aggregation** gruppiert Objekte, Ereignisse und Sachverhalte der Welt in Klassen oder Kategorien. Eine *Kategorie* ist die Extension eines Begriffs. Die **Charakterisierung** beschreibt eine Kategorie, so daß für neue Objekte entschieden werden kann, in welche Kategorie sie gehören. Die intensionale Beschreibung der Kategorie dient also zur Bestimmung der Klassenzugehörigkeit. Ein Objekt wird erkannt als Beispiel eines Begriffs, wenn die Charakterisierung des Begriffs das Objekt abdeckt. Ein *Begriff* ist eine mentale, kognitive Einheit, die sich auf eine Kategorie bezieht. Damit gibt es drei Phänomenbereiche:

Aggregation → Charakterisierung → Klassifikation (Erkennung)

Die Einteilung dient der Strukturierung wissenschaftlicher Arbeit. Die Phänomenbereiche sollen nicht als Phasen eines linearen Ablaufs beim Menschen verstanden werden.

3.1.1 Gründe für die Aggregation

Zunächst könnte man als einen guten Grund dafür, Objekte der Welt zu einer Kategorie zusammenzufassen, angeben, daß sie ein Merkmal gemeinsam haben. Sie sind sich ähnlich. Da aber Merkmale nicht bereits in der Welt vorkommen, sondern ihrerseits gebildet werden, könnten wir umgekehrt für jede Zusammenstellung von Objekten ein Merkmal einführen, das genau für diese Menge gilt. Bei k Objekten gibt es prinzipiell 2^k Mengen von Objekten. Tatsächlich verwenden Menschen aber nicht so viele Kategorien. Es muß also noch zusätzliche Gründe geben, warum Kategorien gebildet werden. Drei Gründe, die in der Literatur diskutiert wurden, werden im folgenden angeführt.

Einige Objekte spielen eine wichtige Rolle für bestimmte Handlungen. Damit begründen die Handlungen einen Bedarf für eine Kategorie. Wenn der Bedarf nur kurzfristig und einmalig ist, so werden Kategorien ad hoc gebildet und danach nicht weiterhin verwendet[2], ansonsten wird die Kategorie konventionalisiert. Quine sah in der Notwendigkeit, etwas vorhersagen zu können, das Motiv für individuelle und gesellschaftliche Kategorienbildung [Quine, 1977]. Ein neuer Begriff wird dann eingeführt, wenn er Objekte klassifizieren kann, deren Verhalten wir vorhersagen wollen. Als Beispiel für eine zunächst unsinnige Kategorie, die aber durch einen bestimmten Handlungszusammenhang sinnvoll werden kann, führen Murphy und Medin gestreifte Objekte mit mehr als einem Bein an, die zwischen 11 und 240 kg wiegen [Murphy und Medin, 1985]. Im Kontext eines Spielfilms, in dem diese Objekte Außerirdische sind, die die Menschheit bedrohen, wird die Kategorie sinnvoll. Es ist dann wichtig zu erkennen, wer dieser Kategorie angehört, wie er sich verhalten wird und wie Menschen sich vor ihm schützen können. Eine andere üblicherweise sinnlose Kategorie besteht aus Primzahlen und Äpfeln. Wenn dies aber die einzigen Gesprächsthemen für die Kollegin Wilma sind, so erhält die Kategorie einen Bezug zu anderen Kategorien (Wilma, Gespräche) und ist nicht mehr absurd [Murphy und Medin, 1985]. Murphy und Medin betonen die Begriffsstruktur, die unterschiedliche Begriffe im Zusammenhang repräsentiert. Erst durch den Zusammenhang wird eine Kategorie oder ein Begriff sinnvoll.

Ein Bedarf an Kategorien wird auch durch ihre Verwendung für die Charakterisierung anderer Kategorien gegeben. Zum Beispiel ist es sinnvoll, die Kategorie *Räder* zu bilden, wenn wir Fahrzeuge definieren wollen. Im Zuge der Charakterisierung von Fahrzeugen entsteht eine neue Begriffsbildungsaufgabe. Es ist einfach praktischer, einen Begriff *Räder* zu haben, als stets die zugehörigen Objekte aufzuzählen: schließlich umfaßt der Begriff eine potentiell unendliche Menge. Nebenbei hebt dieses Beispiel den Zusammenhang von Begriffen hervor: Begriffe werden nicht isoliert voneinander gebildet.

Oft untersuchen psychologische Experimente die Charakterisierung von Kategorien, die von den Psychologen vorgegeben werden. Dabei kann es sich um existierende oder um künstlich gebildete Zusammenstellungen von Objekten handeln. Zu der Kategorie des Belebten (*living thing*) gibt es seit Piaget eine Fülle von Untersuchungen, die verschiedene Charakterisierungsansätze jeweils einer bestimmten Altersstufe zuordnen. Eine Untersuchung von Susan Carey gibt obendrein Hinweise auf die Aggregation von Objekten [Carey, 1985]. In ihrem Exeriment sollten die Kinder zunächst belebte Objekte aufzählen. Das war für fast alle Kinder kein Problem. Man kann annehmen, daß sie diese Kategorie bereits vor dem Experiment kannten. Als sie aber Beispiele unbelebter Objekte anführen sollten, hatten die Kinder Schwierigkeiten. Sie gaben Beispiele für unbelebte Objekte, tote Menschen oder Tiere, Fabelwesen und Abbildungen von Menschen und Tieren (z.B. im Fernsehen) an. Also führten sie unterschiedliche Kategorien an, aus denen sie – möglicherweise

[2] Zu ad hoc Kategorien siehe [Barsalou, 1983]

ad hoc – *Nicht-Belebtes* bildeten. Interessant ist dabei, daß diese neue Kategorie unter verschiedenen Gesichtspunkten in Bezug auf die gegensätzliche Kategorie gebildet wurde. Dies ist ein weiterer Hinweis darauf, daß Kategorien und Begriffe im Zusammenhang gebildet werden.

3.1.2 Probleme der Charakterisierung

Der klassische Ansatz zur Erklärung der Begriffsbildung betrachtet die Charakterisierung als das Finden solcher Merkmale, die alle Beispiele bzw. Instanzen eines Begriffs gemeinsam haben. Ein Begriff ist dann durch eine Menge solcher Merkmale repräsentiert. Daß die Ähnlichkeit von Objekten nicht ausreicht, eine Kategorie zu bilden, haben wir bereits oben festgestellt. Aber auch zur Charakterisierung reichen ähnliche Merkmale nicht aus.

Das erste Problem des klassischen Ansatzes ist die Herkunft der Merkmale. Dimensionen wie Farbe, Größe oder Formen werden selbst erst gebildet, sie sind nicht vorgegeben. Merkmale stammen aus der Wahrnehmung. Land hat gezeigt, daß die Farbwahrnehmung nicht nur auf der Wellenlänge beruht, sondern ebenso auf der Textur des Objekts und der Lichtreflexion [Land, 1983]. Es liegt an dem menschlichen Körper, daß Farben so wahrgenommen werden. Ein Vogel mag Farben anders erfahren. Biologische Untersuchungen können also über einen Aspekt der Herkunft von Merkmalen Auskunft geben: ihre Verankerung in der Wahrnehmung (Stichwort: *symbol grounding*). Sie können jedoch nicht die kulturellen und situationsspezifischen Unterschiede der Wahrnehmung erklären. Lenneberg wies die Abhängigkeit der Farbwahrnehmung von der durch Wörter einer natürlichen Sprache gegebenen Einteilung des Farbspektrums nach [Lenneberg, 1967]. Farben werden als mehr in der Mitte des Bereiches, der durch ihr Wort bezeichnet wird, wiedergeben, als sie wirklich waren. So wurde ein grünliches Blau blauer wahrgenommen, wenn die Sprache kein eigenes Wort für diesen Farbton besitzt. Türkis wird von Menschen, die das Wort türkisïn ihrem aktiven Sprachschatz haben, genauer von Blau abgegrenzt, als von solchen Versuchspersonen, die nur *blau* und *grün* verwenden. Damit werden Unterschiede von Farbtönen, die zum selben Begriff gehören, verringert. Gleichzeitig werden Unterschiede zwischen Farbtönen verschiedener Begriffe verstärkt. Auch der Einfluß der (situationsbedingten) Erwartungen auf die Farbwahrnehmung wurde erwiesen. Die übliche Farbe eines Objektes wird auch dann gesehen, wenn eigentlich eine andere gegeben ist. Es gibt also eine Rückwirkung des sprachlichen und begrifflichen Wissens auf die Wahrnehmung. Insofern erklärt die Verankerung von Merkmalen in der Wahrnehmung wenig.

Das zweite Problem des klassischen Ansatzes ist die Auswahl von Merkmalen. Selbst wenn wir einen Prozeß annähmen, der aus Wahrnehmungen Merkmale formt, so könnten zur Charakterisierung eines Begriffs doch fast unendlich viele Merkmale herangezogen werden. Weitere Einschränkungen sind nötig. Wie schon bei der Kategorienbildung kann auch bei der Charakterisierung die Definition anderer Begriffe zur Auswahl der Merkmale herangezogen werden. Begriffe werden im Zusammenhang definiert. Es werden Merkmale ausgewählt, die solche Begriffe unterscheiden, die nicht verwechselt werden sollen. Die Gegensatz-Beziehung von Begriffen wählt nur die Merkmale zur Charakterisierung aus, die für alle gegensätzlichen Begriffe anwendbar sind und sie unterscheiden. Carey beobachtete außerdem, daß Kinder, wenn sie einmal bestimmte Merkmale dafür benutzten, einen Begriff zu charakterisieren, gegensätzliche Begriffe mit anderen Werten derselben Merkmale definierten [Carey, 1985]. Dies wird *Konsistenz der Charakterisierung* genannt. Eine Folge dieses Prinzips ist, daß Änderungen eines Begriffs Folgen für seine Gegensatz-Begriffe haben. Zusammen mit der Gegensatzrelation zwischen Begriffen hilft die Unterbegriffsrelation bei der Merkmalsauswahl. Voneinander abzugrenzen sind ja nur solche Begriffe, die überhaupt verwechselbar sind. Insbesondere gegensätzliche Unterbegriffe desselben Oberbegriffs werden mit denselben Merkmalen beschrieben.

Susan Carey betont die Abhängigkeit der Begriffsstruktur von dem Wissen eines Menschen [Carey, 1985]. Die Definition des Belebten hängt ab von dem Wissensstand über Biologie. Auch Keil und Kelly zeigen, daß Versuchspersonen mit wenig Wissen über einen Sachbereich eher beschreibende Merkmale auswählen, während zur Verwendung definitorischer Merkmale mehr Wissen nötig ist [Keil und Kelly, 1987]. Die Verschiebung von Beschreibungen zu Definitionen ist damit nur indirekt einer Altersstufe zuzuschreiben – sie ist die Folge des wachsenden Wissens. Kinder wie Laien bevorzugen leicht erkennbare, Fachleute – und für das Alltagswissen sind Erwachsene Fachleute – nutzen gut abgrenzende Merkmale. [Murphy und Medin, 1985] sprechen von einem Netzwerk erklärender Merkmale. Sie geben dafür Beispiele an, daß eine Theorie Merkmale auszuwählen vermag und auch Merkmale korreliert. Biologische Theorien über das Wachstum von Pflanzen geben *besteht_aus_Zellen* und *wächst* den Vorzug vor der Farbangabe. Diese Merkmale hängen zusammen. Sie gelten für alle Pflanzen und werden also auch an z.B. Karotten vererbt. Würde man nun erfahren, daß Karotten gar nicht aus Zellen bestehen, so müßte man den Begriff *Pflanze* ändern. Trifft man hingegen auf blaue Karotten, so sind von dieser Änderung andere Begriffe nicht betroffen. Definitorische Merkmale kann man an dem Ausmaß der Konsequenzen für andere Begriffe erkennen. Definitorische Merkmale charakterisieren Oberbegriffe derart, daß sie an Unterbegriffe weitergegeben werden können.

Das dritte Problem des klassischen Ansatzes besteht in der Begriffsrepräsentation. Eine reine Ansammlung von Merkmalen strukturiert Begriffe nicht. Die Beziehungen zwischen Begriffen ebenso wie die Beziehungen zwischen Merkmalen scheinen aber sehr wichtig zu sein und sollten deshalb repräsentiert werden.

> *In order to characterize knowledge about and use of a concept, we must include all of the relations involving that concept and the other concepts that depend on it.* ([Murphy und Medin, 1985], S. 297)

Begriffliche Gegensätze und Unterbegriffe sollten zusammen mit der Konsistenz ihrer Charakterisierungen und der Vererbung definitorischer Merkmale dargestellt werden. Wir können Merkmale verallgemeinern zu Begriffen, so daß die Relationen zwischen

Merkmalen in derselben Weise behandelt werden wie Begriffsrelationen. Tatsächlich ist es ja nicht einzusehen, warum Zellen mal ein Merkmal sind (wenn wir Pflanzen beschreiben wollen), mal selbst der Begriff sind, der definiert werden soll (wenn wir über Zellen sprechen). Ist die Begriffsrepräsentation nur eine eine Liste von Merkmalen, so hängt sie von dem jeweiligen Gesprächsgegenstand ab. Werden Begriffe jedoch durch ihre Zusammenhänge untereinander dargestellt, so kann – ohne eine Änderung der Repräsentation – auf einen Begriff als Gesprächsgegenstand zugegriffen werden oder als Charakterisierung eines anderen Begriffes. Zum Beispiel können *Karottenfarbe* und *Aprikosenfarbe* als Unterbegriffe von *Modefarbe* genauso genutzt werden wie zur Charakterisierung der jeweiligen Pflanzen. Ein weiterer Vorteil der Vereinheitlichung von Begriffen und Merkmalen besteht in der Änderbarkeit der Begriffsstruktur. Wenn wir über Zellen etwas hinzulernen, ändert sich die Charakterisierung von Pflanzen, die ja aus Zellen bestehen, automatisch – wir müssen keinen zusätzlichen Prozeß annehmen, der dies neue Wissen in den Begriff Pflanze überträgt.

Schließlich soll nicht verschwiegen werden, daß auch diese Sicht auf Begriffe noch nicht alle Probleme löst. Gerade alltägliche Begriffe wie *Tasse* oder *Schuhe* werden auch durch Zusammenhänge zwischen Begriffen noch nicht hinreichend erklärt. Das Wesentliche einer Tasse ist weder ihre Form noch ihre Unterbegriffs-Beziehung zu Behältern, sondern daß wir daraus trinken. Natürlich kann man eine Relation *wird-benutzt-für* einführen, aber das wäre nur ein netter Name. Tatsächlich ist die Tätigkeit des Trinkens selbst das Entscheidende für die Feststellung, ob etwas eine Tasse ist oder nicht. Es sind also nicht nur Beziehungen zu anderen Begriffen, sondern auch zu Handlungen, die Alltagsbegriffe ausmachen.

3.1.3 Beiträge aus dem maschinellen Lernen

Das maschinelle Lernen ist zunächst dem klassischen Ansatz gefolgt und hat den Aggregationsschritt vorausgesetzt. So gibt es eine Fülle von Systemen, die aus vorgegebenen Beispielen für einen Begriff dessen Charakterisierung induzieren. Systeme zum *conceptual clustering* – obwohl auch meist ähnlichkeitsbasiert – beschreiben immerhin den Aggregationsschritt mit. Ein Oberbegriff wird zu einer Hierarchie von Begriffen verfeinert, wobei ähnliche Objekte zusammengruppiert werden. Beispiele sind UNIMEM [Lebowitz, 1987] und COBWEB [Fisher, 1987].

Die Notwendigkeit für komplexere Repräsentationen und das Einbeziehen von Hintergrundwissen wurde von Michalski schon 1983 und von Kodratoff und Ganascia 1986 dargestellt [Michalski, 1983], [Kodratoff und Ganascia, 1986]. Insbesondere die logik-orientierten Ansätze beschäftigen sich mit der Generalisierung aus eingeschränkt prädikatenlogischen Formeln ([Morik, 1987], [Kietz und Wrobel, 1991], [Muggleton und Buntine, 1988], [Rouveirol und Puget, 1990], [De Raedt, 1991]).

Der Bedarf für einen Begriff wird bei CLUSTER/S durch einen Zielgraphen expliziert, der das *conceptual clustering* Verfahren steuert [Michalski und Stepp, 1986]. Stefan Wrobel [Wrobel, 1989] – wie auch schon [Emde et al., 1983] – beschreiben den Bedarf, der durch Ausnahmen einer ansonsten erfolgreichen Regel gegeben ist: sie sollen durch einen neuen Begriff zusammengefaßt werden, auf den eine zusätzliche Prämisse der Regel verweist. Das System KLUSTER beschreibt den Bedarf für einen neuen Begriff aufgrund der Unfähigkeit, mit den vorhandenen Begriffen eine Kategorie zu charakterisieren [Morik und Kietz, 1989], [Kietz und Morik, 1993].

Aktuelle Arbeiten versuchen, die Verankerung der Begriffsbildung in der Welt zu modellieren. So schlägt Stefan Wrobel einen kognitiv motivierten Forschungsrahmen vor, in dem strikt inkrementell gelernt wird [Wrobel, 1991]. Das heißt, die Eingabedaten stellen einen Strom von Informationen dar, der nicht vollständig gespeichert wird. Vielmehr werden die Daten nach und nach strukturiert und diese Strukturierung auf nachfolgende Eingaben angewandt. Revisionen können nicht anhand aller bereits gegebenen Daten überprüft werden.

Die Psychologie liefert in verschiedenen Teilgebieten, besonders Entwicklungspsychologie und kognitive Psychologie, interessante Studien zum menschlichen Lernen von Begriffen demnächst [Barsalou, 1983], [Carey, 1985], [Keil und Kelly, 1987], [Land, 1983], [Lenneberg, 1967], [Rosch, 1978], [Piaget, 1977], [Scholnick, 1983], [Murphy und Medin, 1985].

3.2 Induktion und Abduktion

Der deduktive Schluß ist bisher am längsten und gründlichsten untersucht worden. Im maschinellen Lernen steht der induktive Schluß im Vordergrund. Neuerdings wird auch der abduktive Schluß einbezogen. Wir können die drei Schlüsse folgendermaßen darstellen:

Sei H ein allgemeiner Satz (eine allquantifizierte Aussage oder Hypothese) der Form c_1 & c_2 & ,..., & $c_m \rightarrow c_{m+1}$ oder eine Menge solcher Sätze, die gemeinsam die Hypothese bilden. Wir notieren ein Literal c_i aus den Prämissen mit den Indizes $_1$ bis $_m$ und ein Literal aus der Konklusion mit dem Index $_{m+1}$. Wenn wir die Hypothese in Aussagenlogik¸ff ausdrücken, so gibt es keinen Unterschied zwischen dem Satz und einem Beispiel für den Satz. Wenn wir jedoch in einer (eingeschränkten) Prädikatenlogik formulieren, so kommen in den Literalen Variable $x_1, ..., x_k$ vor. Eine Substitution σ ersetzt Variable durch Konstante. Wenn c_1 etwa $mensch(x)$ ist und $\sigma = \{x/uta\}$, dann ist $c_1\sigma$ das Grundbeispiel $mensch(uta)$. Ein Grundbeispiel (auch: Instanz, Datum oder Fakt genannt) enthält keine Variablen mehr. Die Menge aller Grundbeispiele für Prämissen nennen wir D_m, die Menge aller Grundbeispiele für Konklusionen D_{m+1}. D ist die gesamte Menge von Beispielen $d_1, d_2, ..., d_n$, die sich aus der Vereinigung von D_m und D_{m+1} ergibt.

> $H \cup D_m \vdash D_{m+1}$ ist ein **deduktiver Schluß** und zwar der Modus Ponens.
> Auch $H \vdash D$ ist ein deduktiver Schluß und zwar die Instantiierung.[3]

[3] $(\forall x | p(x)) \vdash p(a)$ ist etwa eine solche Instantiierung, wenn p ein Prädikat, x eine Variable und a ein konstanter Term ist.

Beispiel:

$$\forall x | mensch(x) \rightarrow sterblich(x), mensch(uta) \vdash sterblich(uta)$$

Der induktive Schluß kann nun als die Umkehrung der Instantiierung aufgefaßt werden [Bürckert, 1992].

$D \prec H$ ist ein **induktiver Schluß**.

Beispiel:

$$\begin{array}{ll} D: & mensch(uta), \quad sterblich(uta), \\ & mensch(udo), \quad sterblich(udo), \\ & mensch(uwe), \quad sterblich(uwe) \\ & \prec \\ H: & \forall x | mensch(x) \rightarrow sterblich(x) \end{array}$$

Wenn wir noch eine Theorie als Hintergrundwissen hinzunehmen, so ist der induktive Schluß:

$$T \cup D \prec H,$$

wobei

$$T \not\vdash D_{m+1}, T \cup H \vdash D \mathrm{und} T \cup D \not\vdash \neg H, T \cup H \cup D \not\vdash \Box$$

Das heißt, die Beispiele folgen erst aus der Theorie, wenn der allgemeine Satz H (die Hypothese) hinzugenommen wird, vorher nicht. Die Hypothese ist genereller als die Grundbeispiele, weil diese aus ihr ableitbar sind. Außerdem ist die Hypothese konsistent mit der Theorie und den Beispielen, d.h. aus Theorie und Beispielen wird nicht die Negation der Hypothese abgeleitet bzw. die Theorie, die Hypothese und die Beispiele leiten nicht den Widerspruch (Falsum) ab. Es ist das **induktive Lernproblem**, eine Hypothese für eine Menge von Beispielen (bei gegebener Theorie) zu finden, so daß die angegebenen Bedingungen gelten. Ein wichtiges Teilproblem davon ist zu entscheiden, ob es eine solche Hypothese H gibt. Dieses Problem wird manchmal **Konsistenzproblem** genannt.

Beispiel:

$$\begin{array}{ll} T: & \forall x | mensch(x) \rightarrow säugetier(x), \\ D: & mensch(uta), \quad sterblich(uta), \\ & mensch(udo), \quad sterblich(udo), \\ & mensch(uwe), \quad sterblich(uwe) \\ & \prec \\ H: & \forall x | säugetier(x) \rightarrow sterblich(X) \end{array}$$

Der **abduktive Schluß** schließlich wird sehr unterschiedlich aufgefaßt. Im einfachsten Falle ist er die Umkehrung des Modus Ponens [Bürckert, 1992]. Die Abduktion wird manchmal auch als *hypothetisches Schließen* bezeichnet.

$$H \cup D_{m+1} \succ D_m$$

Beispiel:

$$\begin{aligned} H: &\quad \forall x | mensch(x) \rightarrow sterblich(x), \\ D_{m+1}: &\quad sterblich(uta) \\ \succ &\quad mensch(uta) \end{aligned}$$

Weder der induktive noch der abduktive Schluß sind wahrheitserhaltend. Nur bei dem deduktiven Schluß folgt aus etwas Wahrem immer etwas Wahres. Die folgenden Abschnitte handeln überwiegend von dem induktiven Schluß. Nur das erklärungsbasierte Lernen, das auch dargestellt wird, verwendet ihn nicht zum Lernen.

3.3 Anwendungen maschinellen Lernens

Maschinelles Lernen wird überwiegend eingesetzt, um eine Menge von Regeln aus Daten zu gewinnen oder eine gegebene Regelmenge zu verbessern. Die Regeln werden dann entweder direkt von Menschen verwendet oder einem Expertensystem und damit dessen Benutzer zur Verfügung gestellt. Schon der Einsatz einfacher Lernverfahren führt zu einer erheblichen Verkürzung der Entwicklungszeit einer Wissensbasis. Meist wird anhand einer ausgewählten Teilmenge von klassifizierten Daten (Lernset) eine Menge von Regeln oder ein Entscheidungsbaum induziert. Das Lernergebnis wird dann anhand einer anderen Teilmenge der klassifizierten Daten (Testset) geprüft. Dabei wird der Testset ohne die vorgegebene Klassifikation mit dem Lernergebnis klassifiziert. Wenn die Klassifikation durch das Lernergebnis mit der benutzergegebenen Klassifikation übereinstimmt, ist es korrekt. Wenn nicht, wird noch einmal mit einem anderen Lernset gelernt oder per Hand das Lernergebnis verbessert.

Donald Michie berichtet über erfolgreiche Anwendungen von Lernverfahren, die Entscheidungsbäume induzieren [Michie, 1989]. Dabei muß das Lernergebnis nicht unbedingt von einem Expertensystem genutzt werden. Oft hilft bereits das Ausdrucken des Entscheidungsbaumes als Merkzettel. Die Erstellung des komprimierten Merkzettels ist die Leistung des Lernverfahrens. So führt Michie den Erfolg bei einer Anwendung für die NASA darauf zurück, daß Menschen selten mehr als 3-5 Faktoren auf einmal berücksichtigen können. Falls mehr als 5 Faktoren zu einer Entscheidung beitragen, ist ein induzierter Entscheidungsbaum, der auf der Grundlage aller vorliegender Daten und aller Faktoren gebildet wurde, hilfreich. Insofern hilft das statistisch basierte Lernen bei der Analyse von Daten, deren Ergebnis in eine verständliche, geordnete Form übertragen wird. Diese Analyseleistung war auch bei einem anderen von Michie angeführten Beispiel ausschlaggebend für den Erfolg. Im Bankenbereich der Kreditvergabe werden sicherheitshalber Kredite nicht vergeben, die in einer Grauzone liegen. Mithilfe eines Produktes, das auf ID3 beruht, konnten Daten über zurückgezahlte und nicht zurückgezahlte Kredite analysiert werden. Das Ergebnis strukturiert diese Grauzone, so daß mehr Kredite sicher vergeben werden können. Da das Ergebnis die Faktoren nennt, die ausschlaggebend für eine sichere Kreditvergabe sind, kann die Erwartung für das Kreditvolumen anhand statistischer Kenntnisse aktuell angepaßt werden. Ein Seiteneffekt war, daß der Bank bessere Kundenprofile für ihren Kundendienst zur Verfügung stehen. Schließlich konnte Michie von einer Firma eine schriftliche Bestätigung erhalten, daß die Produktivität einer Fabrik von 83% auf 95% gesteigert werden konnte durch den Einsatz induktiven Lernens[4].

Oft ist eine Induktionskomponente in eine Wissenserwerbsumgebung für eine Expertensystem-Hülle integriert wie zum Beispiel beim System IKEE für die Expertensystem-Hülle TWAICE[5]. Exemplarische Anwendungen verschiedener Lernverfahren werden in dem ESPRIT-Projekt "Machine Learning Toolbox" (P2154) in Zusammenarbeit von Industrieunternehmen, Universitäten und Forschungsinstitutionen untersucht. So wird zum Beispiel das System MOBAL[6] für unterschiedliche Anwendungen erprobt [Morik et al., 1993]. Ein medizinischer Sachbereich wird einerseits mithilfe von benutzergegebenen Regeln dargestellt. Andererseits lernt das System typische Therapieabläufe aus im Krankenhaus gesammelten und von einem Arzt klassifizierten (und bereinigten) Daten. Mithilfe der Konsistenzprüfung von MOBAL werden Abweichungen festgestellt, die dann analysiert werden.[7] Die eigentliche Anwendung des Lernens besteht also darin, eine statistisch basierte und verständlich formulierte Übersicht über Therapieabläufe aus Daten zu erhalten. Eine andere Anwendung des Systems MOBAL entspricht genauer dem klassischen Anwendungsbereich maschinellen Lernens: eine Wissensbasis zur Zugangsberechtigung von Benutzern zu bestimmten Rechnerleistungen soll mithilfe des Systems erstellt werden. Dabei unterstützt das System verschiedene Aufgaben der Modellierung [Morik, 1993], [Fargier, 1991]. Lernverfahren können Regeln aus Daten gewinnen und anhand von Ausnahmen verfeinern. Der Benutzer kann ebenfalls Regeln eingeben. Diese Regeln werden beim Regellernen und Regelverfeinern berücksichtigt. Neben den Lernverfahren verfügt MOBAL aber noch über andere Komponenten, die den Benutzer bei der Modellierung unterstützen. In der Anwendung ist es meist mit einem isolierten Lernverfahren nicht getan! Eine Datenbank mit Verweisen auf Anwendungen verschiedener Lernverfahren wurde vom Turing Institute in Glasgow ebenfalls im Rahmen des ESPRIT-Projektes "Machine Learning Toolbox" erstellt [Morales, 1990].

Wenn in dem Aufbau und der Verfeinerung von Wissensbasen mithilfe maschinellen Lernens bisher auch die meisten Erfahrungen gesammelt wurden, so gibt es doch keinen prinzipiellen Grund, sich darauf zu beschränken. Vielmehr kann jedes System durch Lernfähigkeit verbessert werden. Das Lernen aus Texten wurde mit

[4]Dies entspricht einer Umsatzsteigerung von 10 000 US$ im Jahr.

[5]TWAICE und IKEE sind Entwicklungen der Nixdorf Computer AG, die nunmehr Teil von SNI ist.

[6]MOBAL wurde an der Gesellschaft für Mathematik und Datenverarbeitung in St. Augustin entwickelt.

[7]Dies Anwendung von MOBAL wurde in Zusammenarbeit mit der Foundation of Research and Technology Hellas durchgeführt.

dem wit-System versucht [Reimer und Pohl, 1991]. Weitere Möglichkeiten bestehen in der Robotik ([Dillmann, 1988], [Spandl und Pitschke, 1991], [Zercher, 1991]) und im Bildverstehen [Rieger, 1990]. An Anwendungen bei Systemen qualitativen Schließens arbeitet unter anderem Igor Mozetic [Mozetic, 1990]. In jüngster Zeit hat sich auch das *data mining* als mögliches und wichtiges Anwendungsgebiet gezeigt. Dabei geht es darum, unübersichtliche Datensammlungen nach Regularitäten zu untersuchen. Dies wird bisher vor allem mit statistischen Methoden versucht (Stichwort: Datenanalyse). Maschinelle Lernverfahren, die meist einen statistischen Kern enthalten, gehen in der Aufbereitung ihrer Ergebnisse über rein statistische Verfahren hinaus, indem sie selbst Hypothesen aufstellen und die Ergebnisse in verständlicher Form ausgeben.

4 Lernen als Suche

Mitchell hat Lernen aus Beispielen als Suche beschrieben [Mitchell, 1982]. Beispiele sind in der Lernliteratur einer Kategorie zugeordnete (klassifizierte) Aussagen. Im Gegensatz dazu sind Beobachtungen nicht klassifiziert. Beim Lernen aus Beispielen wurde also die Aggregation bereits vom Benutzer oder einem anderen System vorgenommen. Beim Lernen aus Beobachtungen gehört die Aggregation zur Lernaufgabe. Die Aufgabe, aus Beispielen zu lernen, ist:

Gegeben:
Beschreibungssprache LE für Beispiele,
Hypothesensprache LC für den Begriff,
Menge P positiver Beispiele (Beispiele für den Begriff),
Menge N negativer Beispiele (Nicht-Beispiele für den Begriff),
Abgleichsprädikat, das Beispiele klassifiziert ($covers$)

Ziel: $c \in LC$ mit $\forall p \in P$, $covers(c, p)$ ist wahr
und $\forall n \in N$, $covers(c, n)$ ist falsch

Der Such- bzw. Hypothesenraum für Begriffe ist die Menge aller mithilfe von LC bildbaren Ausdrücke. Das sind alle möglichen Charakterisierungen, für die dann festgestellt werden muß, ob sie alle positiven Beispiele abdecken und kein negatives. Der einfachste Lernalgorithmus ist demnach der **Aufzählungsalgorithmus**: er zählt alle in LC bildbaren Ausdrücke auf (Hypothesengenerierung) und prüft für jeden, welche Beispiele (und Nicht-Beispiele) abgedeckt werden (Hypothesentest). Sobald die Zielbedingung gilt, hält der Algorithmus an.

Der Aufzählungsalgorithmus funktioniert natürlich nur für aufzählbare Sprachen und ist nicht gerade effizient. Ein übliches Verfahren, einen Algorithmus, der Hypothesen generiert und testet, effizienter zu machen, besteht darin, Bedingungen des Testens bereits bei der Generierung zu berücksichtigen. Im Falle des induktiven Lernens wissen wir, daß wir eine Hypothese suchen, die genereller ist als die Beispiele und spezieller als eine sowohl Beispiele als auch Nicht-Beispiele abdeckende Aussage. Wir tun also gut daran, die Hypothesen nach ihrer Allgmeinheit anzuordnen, um dann schrittweise generellere oder speziellere Hypothesen zu generieren. Gehen wir von den Beispielen aus, um schrittweise generellere Hypothesen zu erzeugen bis alle positiven Beispiele abgedeckt werden, spricht man von einem *bottom-up* Verfahren. Gehen wir von einer alles abdeckenden Aussage aus, die wir schrittweise spezialisieren bis sie kein negatives Beispiel mehr abdeckt, spricht man von einem *top-down* Verfahren. Die Suche in einem strukturierten Raum möglicher Hypothesen kann beschnitten werden, was bei der Suche im unstrukturierten Raum nicht möglich ist. Dort müssen ja alle Hypothesen betrachtet werden, weil man keinen Anhaltspunkt hat, wo im Hypothesenraum der Zielbegriff liegen könnte.

Mitchell schlug für Hypothesensprachen eine Halbordnung (*quasi-ordering*) aufgrund der spezieller-als- bzw. genereller-als-Relation vor. Wenn Begriffe mithilfe von Attributwerten charakterisiert werden, die sich in einer Hierarchie entlang dieser Relation partiell anordnen lassen, läßt sich der Suchraum als Kreuzprodukt der geordneten Attributwerte immer (halb-)ordnen.

spezieller-als:

c1 ist spezieller als c2 genau dann, wenn
$\forall e \in LEgilt : covers(c1, e) \rightarrow covers(c2, e)$, d.h.:
$\{e \in LE \mid covers(c2, e)\} \subseteq \{e \in LE \mid covers(c1, e)\}$

c2 ist eine Generalisierung von c1, weil c2 alle Beispiele abdeckt, die c1 auch abdeckt, und zusätzlich vielleicht noch mehr Beispiele. Mit dieser Angabe kann entschieden werden, ob eine Hypothese genereller oder spezieller als eine andere ist. Dies reicht aber noch nicht aus. Wenn wir schrittweise generalisieren bzw. spezialisieren wollen, müssen wir minimal generellere und minimal speziellere Hypothesen zu einer Hypothese finden.

Schrittweises Generalisieren:

$g, g', c \in LC, e \in LE$
$\neg covers(c, e)$ und
g ist genereller als c und
$covers(g, e)$ und
$\neg\exists g' \in LC$, so daß g genereller als g' ist und $covers(g', e)$.

Es wird also eine speziellste Generalisierung g erzeugt: zwischen sie und die bisherige Generalisierung c paßt keine andere Generalisierung g' mehr. Dies ist insbesondere sinnvoll, wenn e ein positives Beispiel ist, das abgedeckt sein soll.

Entsprechend formalisiert Mitchell auch die Spezialisierung.

Schrittweises Spezialisieren:

$s, s', c \in LC, e \in LE$
s ist spezieller als c und

$\neg covers(s, e)$ und
$\neg \exists s' \in LC$, so daß s spezieller ist als s' ist und
$\neg covers(s', e)$.

Es wird also eine generellste Spezialisierung s erzeugt. Dies ist insbesondere sinnvoll, wenn e ein negatives Beispiel ist, das nicht abgedeckt sein soll.

Jetzt lassen sich drei Lernalgorithmen angeben, die alle die Lernaufgabe wie oben angeführt lösen. Es wird dabei angenommen, daß die allgemeinsten Begriffe aus LC alle Beispiele abdecken, die speziellsten alle Beispiele ausschließen. Oft gibt es mehrere Lösungen für ein Lernproblem. Die angeführten Verfahren halten nach der ersten Lösung an. Alternativ könnte auch ein Präferenzkriterium die beste aus allen Lösungen heraussuchen.

Top-down Lernverfahren:

Beginne mit den allgemeinsten bildbaren Hypothesen;
solange noch negative Beispiele abgedeckt werden, wende auf die Hypothese das schrittweise Spezialisieren an;
wenn eine Hypothese kein negatives Beispiel abdeckt, gib diese Hypothese aus und halte an.

Bottom-up Lernverfahren:

Beginne mit den speziellsten bildbaren Hypothesen;
solange noch nicht alle positiven Beispiele abgedeckt werden,
wende das schrittweise Generalisieren an;
wenn eine Hypothese alle positiven Beispiele abdeckt, gib diese Hypothese aus und halte an.

Tom Mitchell führte zusätzlich zu diesen beiden Algorithmen die bi-direktionale Suche im Versionenraum (versions space) ein, die Spezialisierung und Generalisierung kombiniert [Mitchell, 1982]. Es werden gleichzeitig zwei Mengen bearbeitet: die Menge aller aktuellen Generalisierungen und die Menge aller aktuellen Spezialisierungen. Jedes Element dieser Mengen ist möglicherweise die gesuchte Hypothese. Die Mengen enthalten also alternative Hypothesen. Sobald sich die beiden Mengen überschneiden, ist die Lösung eine der Hypothesen aus der Schnittmenge. Weitere Beispiele können diese Schnittmenge auf eine einzige Hypothese reduzieren. Das ist dann die Lösung.

Versionen-Raum Lernverfahren:

Initialisiere die Menge G mit dem generellsten Begriff und die Menge S mit dem speziellsten (z.B. der leeren Menge).
Solange die Mengen G und S disjunkt sind, lies ein Beispiel e ein und
falls e $\in$ N und e von G abgedeckt wird, spezialisiere G bis e nicht mehr abgedeckt wird, alle e $\in$ P bleiben abgedeckt,
falls e $\in$ P und e von S nicht abgedeckt wird, generalisiere S bis e abgedeckt

wird, alle e $\in$ N bleiben ausgeschlossen.
Wenn der Schnitt von G und S nur noch ein Element enhält, Schnitt ausgeben und anhalten!

Die Mengen G und S sind also folgendermaßen definiert:

G: $\{g,$ so daß $\forall p_i \in P, \forall n_i \in N$ gilt
$covers(g, p_i), \neg covers(g, n_i),$
$\neg \exists g'$, das genereller ist als g und konsistent ist mit $e_i\}$

S: $\{s,$ so daß $\forall p_i \in P, \forall n_i \in N$ gilt
$covers(s, p_i), \neg covers(s, n_i),$
$\neg \exists s'$, das spezieller ist als s und konsistent ist mit $e_i\}$

Dabei sind die e_i allgemein Beispiele, p_i und n_i sind die bisher dem System gezeigten positiven und negativen Beispiele.

Unter **Konsistenz** einer Charakterisierung mit einer Menge von Beispielen versteht man die Zielbedingung: alle positiven und keine negativen Beispiele werden abedeckt. Das Konsistenzproblem ist es, herauszufinden, ob es eine konsistente Charakterisierung gibt. Die Konsistenz umfaßt die Vollständigkeit einer Charakterisierung (alle positiven Beispiele werden abgedeckt) und die Korrektheit einer Charakterisierung (keine negativen Beispiele werden abgedeckt). Eine Charakterisierung h ist **vollständig**, wenn sie alle positiven Beispiele abdeckt. Hier:

$$\forall p_i \in P | covers(h, p_i).$$

Im allgemeineren Fall, im Zusammenhang mit Hintergrundwissen T, ergibt sich für eine Charakterisierung H:

$$\forall p_i \in P | T \cup H \vdash p_i.$$

Eine Charakterisierung ist **korrekt**, wenn sie kein negatives Beispiel abdeckt. Hier:

$$\forall n_i \in N | \neg covers(h, n_i).$$

Im allgemeineren Fall, im Zusammenhang mit Hintergrundwissen T, ergibt sich für eine Charakterisierung H:

$$\forall n_i \in N | T \cup H \not\vdash n_i.$$

Ein Beispiel soll den Versionenraum illustrieren. Nehmen wir an, LC enthielte zwei Merkmale mit hierarchisch angeordneten Werten, wobei die allgemeineren Werte oben, die spezielleren unten stehen.

Die Blätter der Merkmalsbäume werden verwendet, um Beispiele, die darüber gelegenen Merkmale, um Charakterisierungen anzugeben. Beispiele sehen dann so aus:

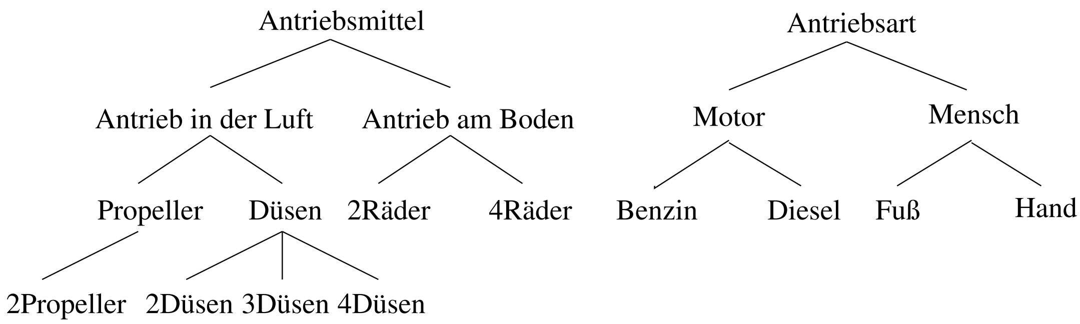

Abbildung 1: Merkmalsbäume

P: { [2Räder,Benzin], [4Räder, Diesel], [2Räder, Fuß]}

N: { [3Düsen, Hand] }

Der geordnete Suchraum stellt alle Kombinationen der beiden Merkmale in der Anordnung von generelleren Charakterisierungen (oben) zu spezielleren (unten) dar. Das Abgleichsprädikat *cover* für die zwei Merkmale ist folgendermaßen:

$$covers([a, b], [c, d])\text{gdw.}covers(a, c) \ \& \ covers(b, d).$$

Dabei sind die Beschreibungen a und b für das Beispiel und c und d für den entstehenden Begriff. Im Suchraum können verschiedene generellere Beschreibungen dieselbe Spezialisierung haben. Da die beiden Merkmalsbäume unterschiedlich tief sind, liegen nicht alle Beispielbeschreibungen (Blätter) auf derselben Ebene des Suchraums. Ein Ausschnitt:

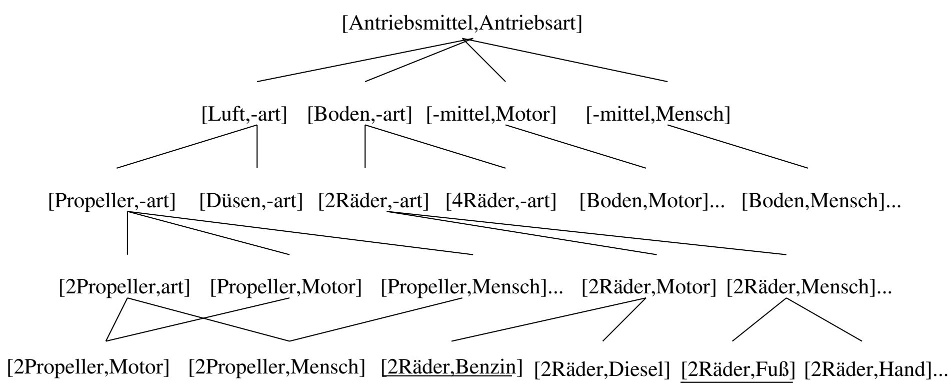

Abbildung 2: Ausschnitt aus dem Versionenraum

Ein Versionenraum-Programm verhält sich etwa so:
?- POSITIVES BEISPIEL?
[2Räder, Benzin]
G: [-mittel, -art] S: [2Räder, Benzin]
BEISPIEL?
[3Düsen, Hand] n
G: [Boden, -art], [-mittel, Motor] S: [2Räder, Benzin]
BEISPIEL?
[4Räder, Diesel] p
G: [Boden, -art], [-mittel, Motor] S: [Boden, Motor]
BEISPIEL?
[2Räder, Fuß] p
G: [Boden, -art], [-mittel, Motor] S: [Boden, -art]
LÖSUNG: [BODEN, -ART]

In diesem Beispiel kann man das Abgleichsprädikat durch die Vorgängerrelation zwischen Knoten im Merkmalsbaum definieren. Der strukturierte Suchraum entsteht dann beim Abgleichen. Obendrein muß das schrittweise Generalisieren und Spezialisieren formuliert werden sowie der globale Ablauf. In Prolog läßt sich das leicht machen.

5 Zwei induktive Lernverfahren

In diesem Abschnitt werden zwei klassische induktive Lernverfahren beschrieben. Das erste ist ein *top-down* Lernen aus Beispielen mit statistischer Merkmalsselektion. Es stellt die Erkennungsfunktion für einen Begriff als Entscheidungsbaum dar. Ein Entscheidungsbaum hat Kanten, an denen Attributwerte stehen, Knoten sind Verzweigungspunkte und Blätter stellen Begriffsnamen dar. Um zu entscheiden, ob ein neues Beispiel zu einem Begriff gehört, wird den Kanten gefolgt, deren Beschriftung einem Attributwert des Beispiels entspricht, bis ein Blatt erreicht ist. Das Beispiel wird dem Begriff zugeordnet, der an dem Blatt angegeben ist. Diese Verfahren heißen *top-down induction of decision trees.* Die bekannteste Realisierung ist ID3 ([Quinlan, 1983]).

Das zweite Verfahren, *conceptual clustering*, lernt aus Beobachtungen, das heißt der Benutzer muß keine Begriffszugehörigkeit angeben. Auch dieses Verfahren enthält eine statistische Bewertungsfunktion. Dabei entsteht eine Hierarchie von Begriffen unter einem Oberbegriff. Es gibt verschiedene conceptual clustering Verfahren. Hier werden zwei vorgestellt: die Stern-Methode (*bottom-up*) und das Verfahren, das in dem System UNIMEM realisiert wurde (*top-down*). Während UNIMEM Attribut-Werte zur Repräsentation verwendet [Lebowitz, 1987], werden bei der Stern-Methode die Begriffe durch eingeschränkte logische Formeln in konjunktiver Normalform charakterisiert. Eine neue Beobachtung wird einem Begriff zugeordnet, indem gezeigt wird, daß die Charakterisierung auf das Beispiel zutrifft.

Die Methode wurde von Michalski entwickelt, eine Realisierung ist CLUSTER [Michalski und Stepp, 1983].

5.1 ID3

Die Lernaufgabe für *top-down* Induktion von Entscheidungsbäumen ist:

Gegeben: eine Menge von Beispielen in einer Attribut-Werte-Repräsentation

Ziel: ein Entscheidungsbaum, der neue Objekte klassifizieren kann

Diese Lernaufgabe ist eine Spezialisierung der oben genannten Beschreibung von Lernen als Suche. Die Beispielbeschreibungssprache ist durch eine Liste von Attributen mit ihren möglichen Werten gegeben. Die Hypothesensprache verwendet dieselben Attribute in einem mächtigeren Formalismus, dem Entscheidungsbaum.

Der Kern des Verfahrens ist die Bewertung des Informationsgewinns eines Attributes für die Klassifikation eines Objektes. Wie gut kann ich ein Objekt klassifizieren, ohne ein Attribut zu kennen? Wie gut kann ich klassifizieren, wenn ich den Wert eines bestimmten Attributes kenne? Welches Attribut bringt den größten Informationsgewinn? Aus der Liste aller Attribute wird zunächst das mit dem größten Informationsgewinn gewählt, angewandt und dann aus der Liste entfernt. Für die verbleibenden Attribute wird dann wieder genauso verfahren, bis schließlich kein weiteres Attribut mehr Informationen liefert – oder die Liste leer ist. Der Informationsgehalt eines Attributs wird durch die Entropie angegeben: $-\sum_{m=1}^{n} p_m log_2 p_m$ bei n verschiedenen Attributwerten und der Wahrscheinlichkeit p_m dafür, daß ein Objekt mit dem m-ten Attributwert zum Zielbegriff gehört. Man vergleicht dann den Informationsgehalt der verschiedenen Attribute mit dem Informationsgehalt der Beispielmenge selbst (also ohne Attribute) und wählt das informativste Attribut oder gar keines. Natürlich kann die Bewertung auch anders gewählt werden, z.B. nach Bayes oder im Sinne der Textkompression (wieviele Zeichen brauche ich bei minimaler Codierung, um etwas auszudrücken). Dieses Forschungsthema soll hier aber nicht behandelt werden.

Der Algorithmus von ID3:

1. Wahl eines Attributs a aus der Attributliste,
   Knoten mit Namen a, Menge C von Beispielen

2. Reduktion der Attributliste um a;
   für alle Attributwerte von a:

   > Kante mit i-tem Wert beschriften;
   > alle Beispiele aus C mit dem i-tem Wert entfernen und in C_i eintragen;
   > C_i als neuen Knoten unter die Kante setzen;

für alle Beispielmengen C_j:

wenn C_j kein Beispiel enthält, schreiben: "Es konnte keine gute Generalisierung gefunden werden!" und anhalten!
wenn alle Beispiele aus Cj demselben Begriff angehören, ist C_j ein Blatt und muß nicht weiter verfeinert werden;
wenn alle Beispielmengen Blätter sind, anhalten!
sonst mit dem Nicht-Blatt wieder zu 1. gehen.

Ein Beispiel soll den Algorithmus verdeutlichen. Als Attributliste für die Beschreibung eines Sachbereichs von fotographischen Aufnahmen für bestimmte Zwecke ist die folgende Attributliste mit den zugehörigen Attributwerten gegeben:

Attributliste:
(Größe {groß, klein},
Filmart {Foto, Dia},
Farbigkeit {s_w, bunt})

Beispiele:
(groß, Dia, s_w, -), (klein, Dia, s_w, -), (groß, Dia, bunt,-),
(groß, Foto, s_w, +), (groß, Foto, bunt,-), (klein, Foto, bunt,-)

Ohne ein Attribut zu kennen, ist die Menge der Beispiele bereits ziemlich geordnet, weil 5 von 6 Beispielen negativ sind und nur eines positiv ist. Die Entropie der Beispielmenge ist 0,649:

$\frac{5}{6}log_2\frac{5}{6} = -0,219$ für negative Beispiele
$\frac{1}{6}log_2\frac{1}{6} = -0,430$ für das positive Beispiel
Summe mit umgekehrtem Vorzeichen: 0,649

Der Informationsgewinn, wenn wir den Wert des Attributs *Größe* kennen, ergibt sich aus der Entropie ohne Attribut minus der Entropie für *Größe*:

groß :
(groß, Dia, s_w, -), (groß, Dia, bunt,-),(groß, Foto, s_w, +), (groß, Foto, bunt,-)
$\frac{3}{4}log_2\frac{3}{4} = -0,311$ für negative Beispiele mit *groß*
$\frac{1}{4}log_2\frac{1}{4} = -0,5$ für das positive Beispiel mit *groß*
Die Summe mit umgekehrtem Vorzeichen ist 0,811.

klein :
(klein, Dia, s_w, -), (klein, Foto, bunt,-)
$\frac{2}{2}log_2\frac{2}{2} = 0$ für die beiden negative Beispiele mit *klein*

Es ergibt sich also

$$\left(\frac{4}{6}0,811\right) + \left(\frac{2}{6}0\right) = 0,541 \quad \text{als Entropie für } \mathit{Größe}.$$

Der Informationsgewinn durch dieses Attribut ergibt sich aus dem Vergleich mit der Entropie ohne ein Attribut: $0,649 - 0,541 = 0,108$

Dieselbe Berechnung muß für alle anderen Attribute durchgeführt werden, damit dann das Attribut mit dem größten Informationsgewinn ausgewählt werden kann. Für *Filmart* ergibt sich:

Foto :

(groß, Foto, s_w, +), (groß, Foto, bunt,-), (klein, Foto, bunt,-)
$\frac{2}{3}log_2\frac{2}{3} = -0,399$ für die 2 negativen Beispiele
$\frac{1}{3}log_2\frac{1}{3} = -0,528$ für das positive Beispiel
Die Summe mit umgekehrtem Vorzeichen ist: 0,927.

Dia :

(groß, Dia, s_w, -), (klein, Dia, s_w, -), (groß, Dia, bunt,-)
$\frac{3}{3}log_2\frac{3}{3} = 0$ für die drei negativen Beispiele.

Es ergibt sich also als Entropie für *Filmart*:

$$\left(\frac{3}{6}0,927\right) + \left(\frac{3}{6}0\right) = 0,463$$

Der Informationsgewinn ist: $0,649 - 0,463 = 0,185$. Damit ist *Filmart* besser geeignet, die Daten zu ordnen als *Größe*.

Wenn *Filmart* ausgewählt wurde, werden zwei Kanten angelegt und mit *Foto* bzw. *Dia* beschriftet. Der unter *Dia* gebildete Knoten enthält nur negativ klassifizierte Beispiele und wird damit zum Blatt. Der unter *Foto* gebildete Knoten muß genauso wie der oberste behandelt werden. Am Ende ergibt sich der in Abbildung 3 abgebildete Entscheidungsbaum.

Ein neues Objekt, zum Beispiel (klein, Foto, s_w), wird nach diesem Entscheidungsbaum nun als neues – hier: positives – Beispiel erkannt. Man kann einen Entscheidungsbaum in eine Menge von Regeln übersetzen. Diese Regelmenge kann in einem Nachbearbeitungsschritt optimiert werden.

5.2 Conceptual Clustering

Conceptual clustering ist aus dem statistischen Verfahren der *cluster* Analyse hervorgegangen. Im Gegensatz zur Statistik, die lediglich numerische Angaben zurückliefert, wird aber beim maschinellen Lernen die Auswertung in Form verständlicher Begriffscharakterisierungen ausgegeben. Zum Beispiel werden die Begriffe in einer Hierarchie angeordnet. Die Lernaufgabe unterscheidet sich von der *top-down* Induktion von Entscheidungsbäumen dadurch, daß der Aggregationsschritt dazugehört. Es wird also nicht aus Beispielen, sondern aus Beobachtungen gelernt.

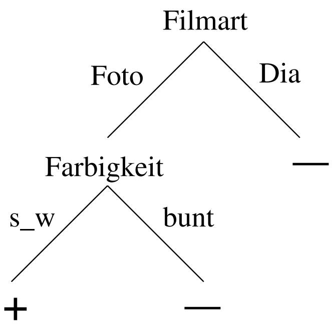

Abbildung 3: Entscheidungsbaum

Gegeben: eine Menge von Beobachtungen

Ziel: eine Hierarchie von Begriffsdefinitionen, die neue Beobachtungen vorhersagen.

Dabei unterscheiden sich die Verfahren, die im folgenden vorgestellt werden, darin, ob sie inkrementell lernen oder nicht. **Inkrementell** ist ein Verfahren, das nicht alle Eingabedaten auf einmal bekommt und dann lernt, sondern jeweils ein zusätzliches Beispiel oder eine neue Beobachtung einliest, daraus lernt, dann das nächste einliest, und so weiter. So war der Versionenraum inkrementell, ID3 hingegen nicht inkrementell. Die Schwierigkeit inkrementellen Lernens besteht darin, daß Entscheidungen bereits getroffen werden, bevor der Rest der Beispiele oder Beobachtungen zur Verfügung steht. Beim Versionenraum machte das keinen Unterschied, weil die Mengen S und G jeweils schrittweise verändert werden und die erreichte Information für weitere Schritte aufbewahren. Bei UNIMEM kann es vorkommen, daß eine Entscheidung aufgrund neuer Beobachtungen rückgängig gemacht werden muß. Zunächst wird das nicht-inkrementelle Stern-Verfahren (*star method*) vorgestellt.

5.2.1 Stern-Verfahren

Die Beschreibungssprache für Beobachtungen ist eine offene Prädikatenlogik (also: ohne Quantoren bzw. nur mit All-Quantoren), wobei Sorten vorgegeben werden. Zum Beispiel gibt es nominale, lineare (geordnete) und hierarchische Wertebereiche für Variablen.

Ein *cluster* ist eine intensional definierte Menge von Beobachtungen, also eine Begriffscharakterisierung. Ein Stern ist eine abgrenzende Beschreibung, also ein elementares *cluster*.

Die Grundidee des Verfahrens ist die Abgrenzung von Mengen von Beobachtungen. Dazu werden zunächst k beliebige Beobachtungen gewählt. Oft ist k=2, so daß ein binärer Baum von Begriffen gebildet wird. Die ausgewählten Beobachtungen werden dann gegeneinander abgegrenzt, d.h. es werden Charakterisierungen gefunden, die die beiden Beobachtungen unterscheiden. Dieser Schritt ist die Stern-Bildung. Aus solchen Charakterisierungen wird eine überschneidungsfreie Abdeckung aller Beobachtungen durch k Begriffe konstruiert. Wählt man recht ähnliche Beobachtungen, so erhält man eher typische Charakterisierungen, deckt aber vielleicht nicht gut genug alles ab. Wählt man sehr unterschiedliche Beobachtungen, deckt man vermutlich viele Beobachtungen gut ab, verpaßt aber vielleicht abgrenzende Merkmale. Ein Bewertungskriterium entscheidet, ob die Menge der Beobachtungen hinlänglich strukturiert ist, oder nicht. Wenn nicht, werden noch einmal andere Ausgangsbeobachtungen gewählt, mit denen die Schritte noch einmal durchlaufen werden. Die gefundenen Begriffe werden weiter verfeinert, indem das Verfahren auf alle von dem jeweiligen Begriff abgedeckten Beobachtungen angewandt wird. Das Verfahren hält an, sobald das Bewertungskriterium erfüllt ist und so viele Ebenen von Begriffen gebildet wurden wie vom Benutzer gefordert. Das Verfahren für eine Ebene von Begriffen noch einmal im Überblick:

Stern-Methode:

1. Wahl von k Ausgangsbeobachtungen
2. Bestimmung des Stern für jede Ausgangsbeobachtung gegen die andere(n)
3. Konstruktion einer disjunktiven Abdeckung
4. Evaluierung:
   wenn das Bewertungskriterium erfüllt ist, alle Charakterisierungen ausgeben und zur nächsten Ebene übergehen.
   wenn das Bewertungskriterium nicht erfüllt ist, für neue Ausgangsbeobachtungen wieder die Schritte 1.-4. ausführen.

Anhand eines einfachen Beispiels soll das Verfahren genauer vorgestellt werden. Dabei nehmen wir Beobachtungen an, die alle durch eine Relation $r(X, Y)$ beschrieben werden, wobei die Wertebereiche für X und Y beide vom Typ *nominal* sind, d.h. die möglichen konstanten Terme werden aufgezählt.

$X : \{Video, 16mm, Super8\}$
$Y : \{Spiel, Trick, Dokumentar\}$

Es gibt also potentiell 9 Beobachtungen, die mit der Relation ausgedrückt werden können. Nehmen wir an, die folgenden vier Beobachtungen wären gegeben:

$e1 : r(Video, Spiel)$
$e2 : r(16mm, Trick)$
$e3 : r(Super8, Dokumentar)$
$e4 : r(Super8, Trick)$

Wie kann man diesen Bereich nun strukturieren?

Wählen wir als Ausgangsbeobachtungen $e1$ und $e4$. Der erste Schritt ist jetzt die Stern-Bildung. Sie soll die Unterschiede zwischen Beobachtungen deutlich machen. Zunächst wird maximal generalisiert, danach soweit als nötig spezialisiert. Ein Stern wird notiert als $G(b_1 \mid b_2)$, wobei b_1 gegen b_2 abgegrenzt wird, d.h. es wird alles notiert, was b_2 nicht hat, aber b_1. Ein Stern besteht aus einer Disjunktion von Merkmalen. Diese Disjunktion ist die maximale Generalisierung, die gerade b_2 noch ausschließt. Es werden bei zwei Ausgangsbeobachtungen zwei Sterne gebildet. Im Beispiel:

$$\begin{aligned} G(e1 \mid e4): \quad & r(\neg Super8, Y) \vee r(X, \neg Trick) \\ & = r(Video \vee 16mm, Y) \vee r(X, Spiel \vee Dokumentar) \end{aligned}$$

Damit sind $e1$, $e2$ und $e3$ abgedeckt und nur $e4$ ist ausgeschlossen. Von den möglichen Beobachtungen sind $e5$, $e6$, $e7$, $e8$ und $e9$ abgedeckt.

$$\begin{aligned} G(e4 \mid e1): \quad & r(\neg Video, Y) \vee (X, \neg Spiel) \\ & = r(16mm \vee Super8, Y) \vee (X, Trick \vee Dokumentar) \end{aligned}$$

Damit sind $e2$, $e3$ und $e4$ abgedeckt, nur $e1$ ist ausgeschlossen.

Nun werden diese beiden Sterne G spezialisiert zu RG. Bisher war die Relation immer nur an einer Argumentstelle eingeschränkt worden. Jetzt wird zu jeder Einschränkung einer Argumentstelle eine passende Einschränkung der anderen gesucht. Dazu werden die Y-Werte zu den im Stern angegebenen X-Werten aufgesammelt. Zu *Video* oder *16mm* gibt es nur *Spiel* oder *Trick*. Entsprechend werden zu den X-Werten die vorkommenden Y-Werte aufgesammelt. Also ergibt sich

$$\begin{aligned} RG(e1 \mid e4): \quad & r(Video \vee 16mm, \ Spiel \vee Trick) \\ & \vee r(Video \vee Super8, \ Spiel \vee Dokumentar) \end{aligned}$$

Damit wird $e1$, $e2$ und $e3$ immer noch abgedeckt. Die Spezialisierung ist nur an den möglichen Beobchtungen zu erkennen. Es ist jetzt $e8$ ausgeschlossen.

$$RG(e4 \mid e1): \quad r(16mm \vee Super8, \ Trick \vee Dokumentar)$$

Damit sind $e2$, $e3$, $e4$ abgedeckt. Von den vorher auch abgedeckten möglichen Beobachtungen sind jetzt $e5$, $e6$, $e7$ und $e9$ nicht mehr abgedeckt.

Bei $G(e4 \mid e1)$ ergibt sich kein Unterschied für $RG(e4 \mid e1)$, ob nun von gegebenen X-Werten aus nach Y-Werten oder von gegebenen Y-Werten nach X-Werten gesucht wird.

Auch die spezialisierten Sterne sind noch nicht überschneidungsfrei für alle (also auch die möglichen) Beobachtungen. Jedes Disjunkt eines Sterns wird mit jedem Disjunkt des anderen Sterns verglichen. So deckt $r(Video \vee 16mm, \ Spiel \vee Trick)$ $e1$, $e2$, $e5$ und $e7$ ab und $r(16mm \vee Super8, \ Trick \vee Dokumentar)$ deckt $e2$, $e3$, $e4$ und $e8$ ab. Sie werden im nächsten Schritt durch Einschränkungen eines Terms disjunkt gemacht. Im Beispiel soll also $e2$ von nur einem Stern abgedeckt werden, muß aus dem anderen ausgeschlossen werden. Dabei gibt es verschiedene

Möglichkeiten, dies zu tun. Hier wird das Verfahren exponentiell. Es wird die erste gefundene Einschränkung gewählt und erst die Qualitätsbewertung der Begriffsdefinition entscheidet, ob diese Möglichkeit gut genug war. Im Beispiel kann $RG(e4 \mid e1)$ eingeschränkt werden zu $r(Super8, Trick \vee Dokumentar)$. Aber auch das zweite Disjunkt von $RG(e1 \mid e4)$, $r(Video \vee Super8, Spiel \vee Dokumentar)$, und $RG(e4 \mid e1)$, $r(16mm \vee Super8, Trick \vee Dokumentar)$, überschneiden sich. Um dies überschneidungsfrei zu bekommen, kann $RG(e4 \mid e1)$ eingeschränkt werden auf: $r(16mm \vee Super8, Trick)$

Damit sind nun *cluster* gebildet und es stehen zwei alternative Begriffsdefinitionen zur Bewertung an:

1. $r(Video \vee 16mm, Spiel \vee Trick)$ für *cluster* um $e1$
   deckt $e1, e2, e5$ und $e7$ ab
   $r(Super8, Trick \vee Dokumentar)$ für *cluster* um $e4$
   deckt $e3$ und $e4$ ab

2. $r(Video \vee Super8, Spiel \vee Dokumentar)$ für *cluster* um $e1$
   deckt $e1, e3, e6$ und $e9$ ab
   $r(16mm \vee Super8, Trick)$ für *cluster* um $e4$
   deckt $e2$ und $e4$ ab

Die Bewertungsfunktion kann vom Benutzer vorgegeben werden. Michalski nennt sie *lexical evaluation function* (LEF). Ein einfaches Maß für ein *cluster* ist:

$$1 - \frac{\text{Anzahl abgedeckter Beobachtungen}}{\text{Anzahl abgedeckter Objekte}}$$

Für die Bewertung der Begriffsqualität werden die Bewertungen zusammengehöriger *cluster* addiert. In unserem Beispiel sind jeweils vom ersten *cluster* um $e1$ zwei tatsächlich beobachtete Objekte abgedeckt und insgesamt vier. Der *cluster* um $e4$ deckt im ersten und zweiten Fall zwei Objekte ab, die beide auch beobachtet wurden. Damit ergibt sich für beide Fälle dieselbe Bewertung von $(1 - 2/4) + (1 - 2/2) = 1/2$. Diese Bewertung erlaubt hier also keine Auswahl zwischen den Alternativen und wir können eine der beiden Definitionen beliebig wählen.

Das Ergebnis einer Iteration ist die Aufteilung des gesamten Bereichs in zwei *cluster*, hier:

$$r(Video \vee 16mm, Spiel \vee Trick) \text{ und } r(Super8, Trick \vee Dokumentar).$$

Diese Begriffe können nun verfeinert werden, indem innerhalb von ihnen wieder *cluster* gebildet werden. In unserem Beispiel macht es wohl keinen Sinn, da die abgedeckten Bereiche bereits sehr klein sind. In realen Anwendungen mit Hunderten von Beobachtungen wird der Algorithmus auf jedes *cluster* erneut angewandt bis eine Mindestanzahl abgedeckter Beobachtungen unterschritten ist.

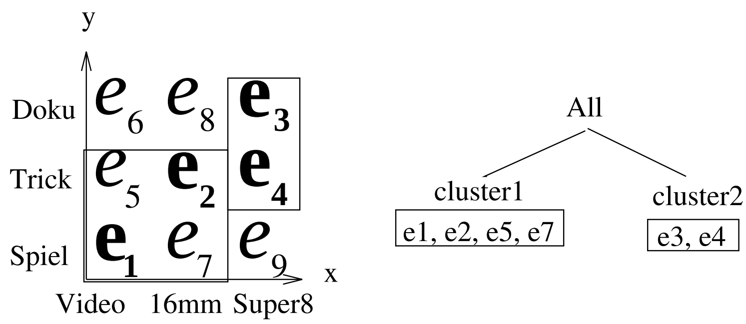

Abbildung 4: Cluster und Beobachtungen

In der graphischen Darstellung (4) sieht man deutlich, daß dies Begriffspaar (*cluster1, cluster2*) nicht alle möglichen Beobachtungen erfaßt. Die beobachteten Ereignisse sind in der Tabell von Beobachtungen fett gedruckt, die möglichen normal. Es kann also nicht vollständig klassifizieren. Bei einem anderen Bewertungsmaß, das die Anzahl aller Objekte mit einbezieht, könnten beide alternativen Begriffspaare abgelehnt werden, weil sie nicht den gesamten Bereich aller möglicher Beobachtungen abdecken. Damit ist die Vorhersagekraft (*predictiveness*) eingeschränkt: nicht alle möglichen Beobachtungen können mit den Begriffen klassifiziert werden. Man könnte also noch einmal andere Ausgangsbeobachtungen wählen, mit denen erst allgemeine Sterne gebildet und dann spezialisiert werden. Zum Beispiel kann man gegensätzlichere Ausgangsbeoachtungen wählen, etwa $e1$ und $e3$. Wenn auch die neuen Begriffspaare nicht genügend Vorhersagekraft haben, kann man die Bewertungen der beiden Iterationen vergleichen. Hat sich die Qualität immerhin verbessert, werden für den nächsten Versuch zentrale Beobachtungen zum Ausgangspunkt gemacht (z.B. $e2$ und $e4$), hat sie sich weiter verschlechtert, gibt es für diese begrenzte Menge von Beobachtungen keine Alternative mehr (andere gegensätzliche Beobachtungspaare, hier: $e6$ und $e9$, wurden nicht beobachtet). Man kann dann annehmen, daß entweder die nicht erfaßten möglichen Beobachtungen tatsächlich nicht vorkommen können oder die Begriffsbildung nicht erfolgreich war.

Das Bewertungsmaß entscheidet, ob das Stern-Verfahren vollständige Begriffsdefinitionen lernt. Die Korrektheit ist im Gegensatz zu Verfahren, die aus Beispielen lernen, schwieriger festzustellen: es gibt keine vorgegebene Einteilung der Beobachtungen in Begriffe, mit der die gefundene Einteilung verglichen werden könnte.

5.2.2 UNIMEM

Wie auch COBWEB [Fisher, 1987] ist UNIMEM [Lebowitz, 1986] ein inkrementelles Lernverfahren, d.h. die Beobachtungen werden nach und nach eingegeben, die Begriffshierarchie an jede neue Beobachtung angepaßt. Die Beobachtungen sind durch ihre Attributwerte beschrieben. Das Lernergebnis ist eine Begriffshierarchie, bei der die Knoten Begriffe und die Kanten Unterbegriffsrelationen darstellen. Ein Begriff ist durch seine Unterbegriffs- und Oberbegriffsrelationen zu anderen Begriffen, seine Merkmale und alle ihm zugeordneten Beobachtungen repräsentiert. Übergeordnete (das sind allgemeinere) Begriffe vererben ihre Merkmale an untergeordnete, also speziellere Begriffe.

Eine Beobachtung wird dem speziellsten Begriff zugeordnet, mit dem sie genügend Merkmale gemeinsam hat. Ein Parameter legt fest, was als *genügend* zu werten ist (Gesamtähnlichkeit). Dieser Parameter entspricht der lexikalischen Bewertungsfunktion der Stern-Methode, die die Güte eines gebildeten Begriffs bewertet. Da UNIMEM aber nicht über alle Informationen auf einmal verfügt, kann der Begriff nicht danach beurteilt werden, wie gut er von allen anderen abgegrenzt ist. Ein weiterer Parameter legt fest, welche Attributwerte einander ähnlich sind (Distanzmaß). So müssen Beobachtungen nicht genau den gleichen Attributwert haben, damit man von ihnen sagt, sie hätten das entsprechende Merkmal gemeinsam. Dies ist insbesondere bei numerischen Werten sinnvoll: wenn die Werte zweier Beobachtungen in einem Intervall liegen, so gelten sie als gleiches Merkmal. Bei der Stern-Methode wurde derselbe Effekt durch die definierten Wertebereiche und das Bilden von Intervallen erreicht.

Wenn ausreichend viele Beobachtungen B_i, die einem Begriff C zugeordnet sind, gemeinsame Merkmale haben, die die anderen demselben Begriff zugeordneten Beobachtungen nicht haben, so wird ein neuer Unterbegriff von C für die Beobachtungen B_i gebildet. Ein Parameter legt fest, wann ein neuer Knoten eingeführt werden soll (Unterscheidbarkeitsmaß). Dies entspricht dem Verfeinern eines *clusters* bei der Stern-Methode. Dort gab es nur die Begriffsbewertung. Hier gibt es zwei Kriterien: die Gesamtähnlichkeit für die Klassifikation und das Unterscheidbarkeitsmaß für die Verfeinerung eines Begriffs.

Nach dieser allgemeinen Beschreibung des Verfahrens wird es nun noch einmal genauer vorgestellt. Das Verfahren besteht aus drei Prozeduren: Suche vom Wurzelknoten aus nach dem speziellsten Knoten für eine Beobachtung anhand aller ihrer Merkmale, Einordnung der Beobachtung in einen Knoten und Bewertung der Merkmale während der Suche nach ihrer Klassifikations verläßlichkeit.

Suche nach dem speziellsten Knoten:

für jeden Attributwert:

> stelle die Distanz fest zwischen dem Wert der Beobachtung und dem Wert, der als Definition bei dem Knoten angegeben ist; wenn die Distanz laut Distanzmaß klein ist, so gilt der Attributwert als *erklärt*, sonst ist er *unerklärt*.

Wenn die Summe der Distanzen größer ist als die geforderte Gesamt-Ähnlichkeit, paßt die Beobachtung nicht in diesen Knoten und auch in keinen seiner Unterknoten, weshalb der Nachbarknoten als nächster betrachtet wird;

Wenn die Summe der Distanzen kleiner ist als die geforderte Gesamt- Ähnlichkeit,

werden als nächstes alle unerklärten Merkmale bei einem Unterknoten untersucht;

Sonst:

es gibt keinen passenden Knoten, also wird die Beobachtung dem Wurzelknoten zugeordnet.

Einordnung einer Beobachtung in einen Knoten C:
alle erklärten Merkmale werden aus der Beobachtungsbeschreibung gelöscht, alle unerklärten Merkmale bleiben Merkmale der Beobachtung;
Wenn die unerklärten Merkmale nicht mit unerklärten Merkmalen anderer Beobachtungen aus C übereinstimmen,

wird die Beobachtung dem Knoten hinzugefügt;

Wenn laut Unterscheidbarkeitsmaß genügend unerklärte Merkmale für eine Menge von Beobachtungen B_i übereinstimmen,

wird ein neuer Knoten D als Unterknoten von C eingeführt, der die übereinstimmenden unerklärten Merkmale als Definition bekommt;
aus den Beschreibungen der B_i werden die ihnen gemeinsamen Merkmale gelöscht;
die B_i werden dem Knoten D hinzugefügt.

Bewertung der Merkmale:
Für jedes Attribut einer Beobachtung, das auch bei dem Knoten vorkommt:

Wenn der Attributwert der Beobachtung dem des Knoten entspricht, wird die Verläßlichkeit des Merkmals beim Knoten erhöht;
Wenn der Attributwert der Beobachtung dem des Knoten direkt widerspricht, wird die Verläßlichkeit des Merkmals beim Knoten gesenkt;

Für jedes Attribut eines Knoten:

Wenn die Verläßlichkeit unter einen Schwellwert (Verläßlichkeitsforderung) gesunken ist, wird das Attribut aus der Begriffsdefinition gelöscht;

> Wenn die Verläßlichkeit über einen Schwellwert gestiegen ist (Verläßlichkeitszusicherung), wird das Attribut unveränderlich durch weitere Beobachtungen.

Die Verläßlichkeit eines Merkmals gibt an, wie stark der betreffende Attributwert definierend ist. Es wird nicht gefordert, daß alle Beobachtungen eines Begriffs für ein definierendes Attribut denselben Wert haben. Es kann also Ausnahmen geben. Aber Merkmale, die bei einigen Beobachtungen (zufällig) vorkommen, können nicht definierend sein. Der untere Schwellwert (Verläßlichkeitsforderung) erlaubt ein Spektrum von Abweichungen. Da das Verfahren inkrementell ist, muß es die Möglichkeit geben, ein Merkmal wieder zu löschen. Es können Knoten entstehen, die nur die Eigenarten der ersten wenigen Beobachtungen wiedergeben. Diese müssen wieder entfernt werden, wenn alle ihre Merkmale unter die Verläßlichkeitsforderung gesunken sind. Nur aufgrund der Möglichkeit, Begriffsdefinitionen auch zu revidieren, ist das Verfahren nicht vollständig abhängig von der Reihenfolge der Beispiele.

Der obere Schwellwert legt fest, wann Abweichungen nichts mehr ändern sollen. Damit dieser Wert vernünftig bestimmt werden kann, muß man vorher wissen, wieviele Beobachtungen ungefähr in das System eingegeben werden. Man kann die Verläßlichkeitszusicherung dafür verwenden, das System nach einer Trainingsphase stabil zu halten.

Ein Beispiel soll das Verfahren verdeutlichen. Nehmen wir an, wir hätten die folgenden Daten (5):

Beob.\Attr.	Höhe	Material	Henkel	Stiel	Untertasse	Außenform
Tasse	flach	Porzellan	mit	ohne	mit	rund
Suppentasse	flach	Porzelan	ohne	ohne	mit	rund
Becher	hoch	Keramik	mit	ohne	ohne	eckig
Wasserglas	hoch	Glas	-	ohne	ohne	eckig
Weinglas	-	Glas	ohne	mit	ohne	-
Teeglas	hoch	Glas	mit	ohne	ohne	eckig
Plastikbecher	-	Plastik	-	ohne	ohne	eckig
Sektglas	hoch	Glas	ohne	mit	ohne	spitz
Cocktailglas	flach	Glas	ohne	-	ohne	-

Abbildung 5: Beobachtungen

Nehmen wir weiterhin an, bisher hätte UNIMEM aufgrund der ersten 7 Beobachtungen die Begriffshierarchie von Abbildung 6 erzeugt.

Wenn nun das *Sektglas* neu eingeordnet werden soll, so werden zunächst die beiden Merkmale *Material-Glas* und *Untertasse-ohne* durch den Knoten G1 erklärt. Die

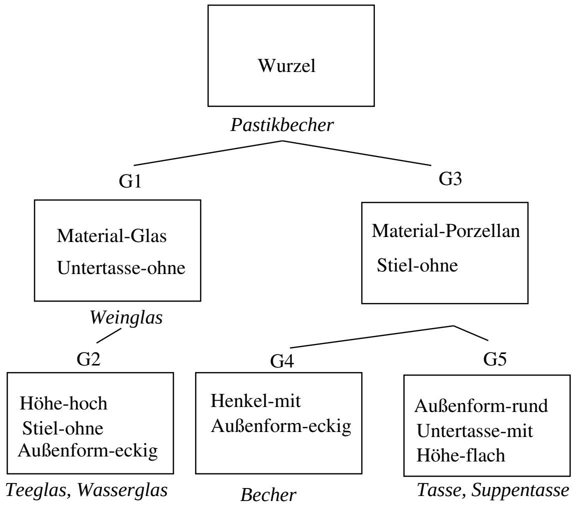

Abbildung 6: Begriffshierarchie

beiden Merkmale *Henkel-ohne* und *Stiel-mit* haben *Weinglas* und *Sektglas* gemeinsam. Wenn das Unterscheidbarkeitsmaß für die Einführung eines neuen Unterknoten bei 1/2 liegt, so wird nun ein neuer Unterbegriff G6 unter G1 gehängt, dem *Weinglas* und *Sektglas* zugeordnet sind. Diese beiden Beobachtungen haben mindestens 2 von 4 noch unerklärten Merkmalen gemeinsam. Es kommt auf das Distanzmaß an, wie mit nicht angegebenen Attributwerten verfahren wird. Wenn man festlegt, daß ein nicht angegebener Wert mit jedem möglichen Wert gleich ist, so hätten *Wein-* und *Sektglas* sogar 4 von 4 unerklärten Merkmalen gemeinsam. Die letzte Beobachtung, das *Cocktailglas* wird ebenfalls G6 zugeordnet. Falls neue Beobachtungen eingegeben werden, die etwas mit *Plastikbecher* gemeinsam haben, wird ein dritter Unterknoten unter den Wurzelknoten eingefügt, der etwa als definierendes Merkmal *Material-Plastik* haben könnte.

UNIMEM gruppiert Beobachtungen zu *clusters* zusammen, die Merkmale gemeinsam haben, und stellt Unterbegriffsrelationen zwischen solchen Gruppen von Beobachtungen auf. UNIMEM kann nicht garantieren, daß es eine korrekte und vollständige Begriffshierarchie lernt. Zunächst bestimmt die Reihenfolge der Beispiele, welche Knoten mit welchen Merkmalen eingerichtet werden. Dann hängt es von den konkreten Parameterbelegungen ab, wie gut die bei fast allen Beobachtungen auftretenden Merkmale in der Begriffshierarchie nach oben gelangen.

6 Deduktives Lernen

Für erklärungsbasiertes Lernen ist die Lernaufgabe nicht, Beispiele oder Beobachtungen zu einer Begriffsdefinition zu verallgemeinern, sondern eine Begriffsdefinition für eine Anwendung zu operationalisieren. Wenn eine Begriffsdefinition in einer Terminologie vorliegt, die erst mühsam aus der von der Anwendung vorgegebenen Terminologie gewonnen werden muß, ist die Neudefinition des Begriffs in Anwendungstermini eine Operationalisierung.

Gegeben:

Zielbegriff mit einer Definition
Übungsbeispiel: positives Beispiel für den Zielbegriff
Sachbereichstheorie
Operationalitätskriterium: ein Prädikat, das entscheidet, welche Termini zur Neudefinition des Zielbegriffs herangezogen werden dürfen.

Ziel:

Eine Definition des Zielbegriffs, die dem Operationalitätskriterium gehorcht.

Die Idee dabei ist, daß eine Lösung (Übungsbeispiel) anhand des Wissens (Sachbereichstheorie) nachvollzogen wird. Dabei wird die Sachbereichstheorie mit den Angaben zum Beispiel in Verbindung gebracht. Aus dieser Verbindung werden dann die operationalen Bestandteile herausgezogen und für zukünftige Beispiele zur Klassifikation genutzt. Es handelt sich also um ein sicheres Lernverfahren: es wird deduziert, daß das Übungsbeispiel von dem Zielbegriff abgedeckt wird. In den Deduktionsschritten werden Variable für Konstante eingesetzt, so daß die Schritte für zukünftige Beispiele direkt genutzt werden können. Der operationale Begriff ist eine Spezialisierung. Da die Sachbereichstheorie erhalten bleibt, können aber auch alle Beispiele, die vor dem Lernen klassifizierbar waren, weiterhin klassifiziert werden.

Das einfache Verfahren von Mitchell verwendet einen Theorembeweiser, um das Übungsbeispiel aus der Sachbereichstheorie abzuleiten [Mitchell, 1985]. In dem Beweispfad werden dann mithilfe der Substitutionen Variablen eingeführt. Die Termini, in denen das Übungsbeispiel beschrieben ist, werden als operational definiert. Die Blätter des Beweisbaumes ergeben dann den operationalen Begriff. Sein Verfahren entspricht der partiellen Evaluation der logischen Programmierung. Das Übungsbeispiel wird lediglich zur Fokussierung auf relevante Beweispfade benutzt [van Harmelen und Bundy, 1988].

Die drei in der Literatur immer wieder angeführten Beispiele für dieses Verfahren betreffen Mord und Selbstmord, Tassen sowie die Stapelbarkeit von Objekten. Letzteres wird hier vorgeführt:

Zielbegriff:

$leichter(X, Y) \rightarrow stapelbar(X, Y)$

Übungsbeispiel:

$auf(obj1, obj2)$, $isa(obj1, kiste)$, $isa(obj2, tisch)$,
$farbe(obj1, rot)$, $farbe(obj2, blau)$,
$volumen(obj1, 1)$, $dichte(obj1, 0.1)$

Sachbereichstheorie:

$volumen(P, V)$ & $dichte(P, D) \rightarrow gewicht(P, V * D)$
$gewicht(P, W1)$ & $gewicht(Q, W2)$ & $W1 < W2 \rightarrow leichter(P, Q)$
$isa(P, tisch) \rightarrow gewicht(P, 5)$
$0.1 < 5$

Operationalitätskriterium:

Nur $volumen$, $dichte$, $<$ und isa sind operational.

Mithilfe der Sachbereichstheorie kann nachgewiesen werden, daß die Kiste auf den Tisch stapelbar ist. Der Beweis wird für die künftige Nutzung durch andere Beispiele verbessert, indem Variablen gemäß der Substitutionen, die im Beweis verwendet wurden, anstelle der Konstanten gesetzt werden (7).

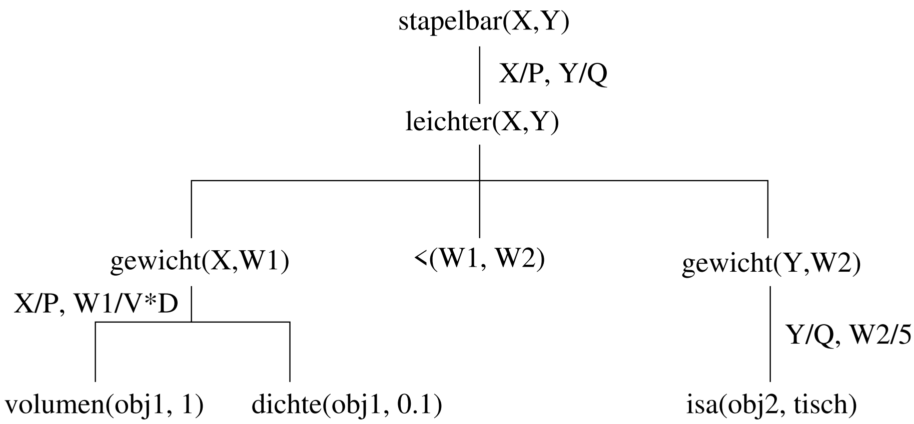

Abbildung 7: Verallgemeinerung des Beweisbaums

Die Substitutionen sind in dem Beweisbaum angegeben.[8] Als operationale Begriffsdefinition ergibt sich:

$$volumen(X, V) \ \& \ dichte(X, D) \ \& \ V * D < 5 \ \& \ isa(Y, tisch) \rightarrow stapelbar(X, Y)$$

Dieses Verfahren kann leicht in Prolog programmiert werden, wobei man Prolog als Theorembeweiser benutzt, der gleich die Substitutionen mitliefert. Falls die Sachbereichstheorie unübersichtlich ist und man stets Anwendungen einer speziellen Form hat, führt die operationale Definition zu einer Performanzsteigerung des Klassifikationssystems. In anderen Fällen jedoch nicht!

An dem Beispiel sind die Schwächen des Verfahrens gut zu erkennen. Das Operationalitätskriterium ist hier einfach eine Aufzählung von Prädikaten. In echten

[8]Für den Schritt der Variabilisierung wird das Mord- ($tötet(X, Y)$) bzw. Selbstmordbeispiel ($tötet(X, X)$) oft angeführt.

Anwendungen kann ein reicheres Kriterium nötig sein, das sich dann nicht mehr so einfach abprüfen läßt [De Jong und Mooney, 1986].

Eigenarten des Übungsbeispiels, die vielleicht nicht immer in der Anwendung vorkommen, geraten genauso in die neue Begriffsdefinition wie die für die Anwendung wichtigen Eigenschaften. So ist hier der Gewichtsvergleich zwischen Tischen und allen anderen Objekten in der Begriffsdefinition enthalten. Das ist sinnvoll, wenn in der Anwendung grundsätzlich nur auf Tische etwas gestellt werden soll. Wenn aber die Operationalisierung der Definition darin bestehen sollte, daß das Prädikat *leichter* durch die Prädikate $volumen, dichte$ und $<$ ersetzt wird, so ist der neue Begriff zu speziell geworden. Wir müßten dann ein anderes Übungsbeispiel wählen, in dem auch das Gewicht des zweiten Objekts berechnet wird, so daß ein symmetrischer Beweisbaum entsteht. Die Wahl des Übungsbeispiels ist also entscheidend. Der neue Begriff kann auch deshalb zu speziell definiert sein, weil Prädikate nicht generalisiert werden [De Jong und Mooney, 1986].

Neuere Arbeiten beschäftigen sich damit, unter welchen Umständen die partielle Evaluierung sinnvoll ist. So ergibt sich nicht immer eine Performanzsteigerung, wenn eine neue, operationale Regel eingeführt wird: Es können so viele neue Regeln sein, daß die Auswahl der anwendbaren Regel länger dauert, als der Beweis gedauert hätte. Bei dem Verfahren von Henrik Boström wird die gesamte Wissensbasis auf ein Problem ausgerichtet. Im Gegensatz zur partiellen Evaluierung wächst bei ihm die Anzahl der Regeln nur linear mit der Anzahl der Beweisschritte im Übungsbeispiel [Boström, 1992]. Auch Peter Clark und Rob Holte verbessern das erklärungsbasierte Lernen. Sie führen eine *lazy evaluation* ein [Clark und Holte, 1992].

7 Logik-orientiertes induktives Lernen

Wie oben schon dargestellt, ist mit Lernen meist ein induktiver Schluß gemeint. Diesen induktiven Schluß für die Prädikatenlogik konstruktiv zu formalisieren, so daß für eine gegebene Menge von Daten (und eine Theorie) die speziellste (oder generellste) Verallgemeinerung gefunden werden kann, ist in jüngster Zeit ein wieder lebhaft diskutiertes Thema geworden. Frühe Ergebnisse von [Plotkin, 1971] waren wenig ermutigend: im allgemeinen Fall ist in der Prädikatenlogik erster Stufe nicht entscheidbar, ob die speziellste mit Hintergrundwissen und Beispielen konsistente und gemäß einer Interessantheitsordnung minimale Generalisierung gefunden wird. Inzwischen ist dieser Satz reformuliert worden: die Prädikatenlogik muß eingeschränkt werden, damit eine speziellste Generalisierung gefunden werden kann. Die aktuellen Ansätze unterscheiden sich zum einen darin, wie die Generalisierung definiert wird, zum anderen in den konkreten Einschränkungen der Prädikatenlogik. Es ist hier nicht der Platz gegeben, alle Ansätze in einheitlicher Weise vorzustellen. Einen Vergleich von Verfahren für speziellste Generalisierungen bietet [Kietz, 1993a]. Es wird aber die Generalisierung mit und ohne Bezug zu Hintergrundwissen behandelt und ein Verfahren kurz vorgestellt.

7.1 Lernen in Prädikatenlogik

Bei den logik-basierten Verfahren geht es darum, genau anzugeben, wann ein Literal oder eine Klausel eine Generalisierung eines anderen Literals bzw. einer anderen Klausel darstellt. Wenn man die Generalisierungsbeziehung formalisieren kann, dann kann man hoffentlich auch ein Verfahren finden, das zu gegebenen Literalen bzw. Klauseln eine Generalisierung konstruiert. Und das wäre dann ein induktiver Schluß.

Eine Möglichkeit, die Generalisierung zu beschreiben verwendet die Implikation. Eine Klausel C1 ist genereller als eine andere, C2, wenn C1 $\rightarrow$ C2 gilt. Um dann eine Generalisierung zu finden, müssen wir die Klausel finden, die die gegebenen Beispiele impliziert. Dies ist schwierig, weil es darauf hinausläuft, die Implikation oder logische Folgerung zwischen Klauseln als Grundlage zu nehmen, die nicht im allgemeinen Fall entscheidbar ist. Deshalb wird die schwächere Subsumptionsbeziehung bevorzugt. Eine Klausel C1 ist genereller als eine andere Klausel C2, wenn gilt: C1 subsumiert C2. Bei Literalen ist das einfach. Ein Literal L1 subsumiert ein anderes Literal L2, wenn es eine Substitution σ gibt, so daß $L1\sigma = L2$. Damit wird Substitution zur Grundlage der Formalisierung von Induktion.

Bei Klauseln ist es etwas schwieriger. Wir können uns Generalisierung für Klauseln einmal (semantisch) an den Objekten (Daten, Beispielen) einer logischen Struktur deutlich machen. Eine Klausel C1 ist genereller als eine andere Klausel C2 (geschrieben: $C1 \geq C2$), wenn sie mehr Objekte abdeckt. So ist zum Beispiel

$$tier(X) \rightarrow s\ddot{a}ugetier(X) \geq tier(rex) \rightarrow s\ddot{a}ugetier(rex) \text{ und}$$
$$tier(X) \rightarrow s\ddot{a}ugetier(X) \geq tier(X) \,\&\, im\_haus(X) \rightarrow s\ddot{a}ugetier(X).$$

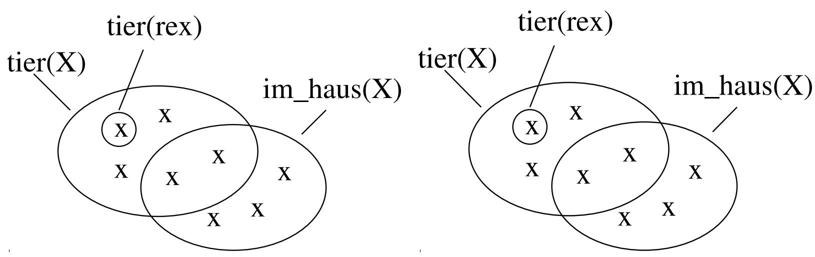

Abbildung 8: Objektmengen

Die Säugetiere umfassen einmal nur *rex* (und vielleicht noch andere Objekte), einmal mindestens die Schnittmenge der beiden Objektmengen, schließlich sogar mindestens alle Tiere. An dem Beispiel ist auch deutlich zu sehen, daß neben der Subsumption, auch die logische Folgerung gilt. Wenn C1 genereller ist als C2, so gilt $C1 \models C2$.

Zum anderen können wir uns die Generalisierung aber auch (syntaktisch) an der Gestalt der Klauseln klarmachen. Wenn wir sie als Mengen schreiben, so ist die generellere Klausel eine Teilmenge der spezielleren.

$$\{\neg tier(X), \neg im\_haus(X), s\ddot{a}ugetier(X)\} \supseteq \{\neg tier(X), s\ddot{a}ugetier(X)\}.$$

Natürlich müssen die Terme so substituiert werden, daß es paßt.

$C1 \geq C2$ gdw. $C2 \supseteq C1\sigma$.

Die Faustregel lautet: je mehr Literale eine Klausel hat, desto spezieller ist sie. Wenn aber durch eine Theorie gegeben ist, daß einige Literale gleichbedeutend sind mit einem anderen Literal, so hilft das einfache Abzählen nichts. Wenn für alle Tiere bekannt ist, daß sie sterblich sind, so wird eine der oben angeführten Klauseln über Tiere nicht spezieller, wenn $sterblich(X)$ hinzugefügt wird. Es ist einfach redundant. Diesen Gedanken werden wir in den nächsten beiden Abschnitten vertiefen, bevor wir dann ein Lernverfahren vorstellen, das darauf aufbaut.

7.1.1 Plotkins Ansatz

Gordon Plotkin stellt ein Verfahren vor, wie in der uneingeschränkten Prädikatenlogik erster Stufe die speziellste Generalisierung (*least general generalization* – lgg oder *most specific generalization* – MSG) gefunden werden kann [Plotkin, 1970]. Dabei stellt er zunächst ein Verfahren vor, das für Literale und Terme (zusammengefaßt unter der Bezeichnung *Wort*) eine speziellste Generalisierung findet.

$W1 \geq W2$, wenn $W1\sigma = W2$.

Zum Beispiel ist

$$p(X, X, g(g(Y))) \geq p(l(3), l(3), f(g(X))), \text{ mit } \sigma = \{X/l(3), Y/X\}.$$

Wieder ist $\geq$ (wie schon bei Lernen als Suche besprochen) eine Halbordnung (also reflexiv und transitiv).

> Für eine Menge von Wörtern K ist W die speziellste Generalisierung, gdw.:
> Für jedes Wort V aus K gilt $W \geq V$ und
> wenn für jedes V aus K $W1 \geq V$, dann gilt auch $W1 \geq W$.

Mit der zweiten Bedingung wird gewährleistet, daß es sich bei W um die speziellste Generalisierung handelt. Wenn obendrein $W \geq W1$ gilt, dann sind W1 und W äquivalent. Plotkins Verfahren für die Generalisierung von Wörtern wählt zunächst solche Wörter aus, die dasselbe Prädikatensymbol mit demselben Vorzeichen (negiert oder nicht negiert) haben. Dann muß er eine Substitution für die Terme finden, die an derselben Stelle des Prädikats auftreten, aber verschieden sind. Er wählt eine

sonst nicht vorkommende Variable, setzt sie für die Terme ein und schreibt die Substitutionen in die Substitutionslisten der Wörter. Dies wird für alle verschiedenen Terme an selber Argumentstelle so gemacht. Wenn es keine solchen Terme (mehr) gibt, so ist das durch die Substitutionen entstandene Wort die speziellste Generalisierung der Wörter.
Beispiel:

$$V1 : p(f(a(), g(Y)), X, g(Y))$$
$$V2 : p(h(a(), g(X)), X, g(X))$$

Y und X haben nehmen dieselbe Argumentstelle ein und werden durch die neue Variable Z ersetzt. Das ergibt

$$V1 : p(f(a(), g(Z)), X, g(Z))$$
$$V2 : p(h(a(), g(Z)), X, g(Z))$$

und die Substitutionsliste für $V1$ ist $\{Z/Y\}$, die für $V2$ ist $\{Z/X\}$.

Als nächstes sind $f(a(), g(Z))$ und $h(a(), g(Z))$ verschiedene Terme an derselben Argumentstelle. Sie werden durch eine neue Variable U ersetzt.

$V1 = V2 = p(U, X, g(Z))$ mit den Substitutionslisten:
$\sigma 1 = \{U/f(a(), g(Z)), Z/Y\}$ für $V1$ und
$\sigma 2 = \{U/h(a(), g(Z)), Z/X\}$ für $V2$.
$p(U, X, g(Z))$ ist die speziellste Generalisierung für $V1$ und $V2$.

Durch Anwendung der Substitutionen auf die Generalisierung erhält man wieder $V1$ respektive $V2$.

Um dieses Verfahren auf Klauseln übertragen zu können, müssen zunächst alle Literale miteinander kombiniert werden. So macht man aus einer Menge eine äquivalente Liste. Wenn etwa die Klausel C1 die Literale L11, L12, L13 enthält und die Klausel C2 die Literale L21, L22, L23, so erhält man die Listen

$$C1 : [L11,\ L11,\ L11,\ L12,\ L12,\ L12,\ L13,\ L13,\ L13]$$
$$C2 : [L21,\ L22,\ L23,\ L21,\ L22,\ L23,\ L21,\ L22,\ L23]$$

Diese Listen können nun einfach elementweise gegeneinander abgeglichen werden.

Dabei kann man alle übereinander stehenden Literalpaare streichen, die nicht dasselbe Prädikatssymbol haben. Für die resultierenden Listen findet Plotkin Generalisierungen mithilfe der θ-Subsumption.

Plotkins Übertragung des Verfahrens für Wörter auf Klauseln ist einfach: er formt Klauseln so um, daß neue Funktionssymbole anstelle der Prädikatssymbole stehen, kann die Prädikatensymbole dann weglassen und verfährt mit den Funktionen wie oben beschrieben. Man behandelt die Menge von Literalen einer Klausel einfach als Terme eines Prädikats!

Die Menge der Literale in den Klauseln werden mit zwei Indizes versehen, eines für die Klausel (i, j), eines für die Literale selbst $(1, .., l, .., n)$. Jedes Literal L_{jl} hat ja die Form $(\pm)p_l(t_1, ..., t_k)$.
Bei n Literalen einer Klausel wird ein neues, n-stelliges Prädikat q konstruiert, das für jedes k-stellige Prädikat der Klausel eine neue k-stellige Funktion als Argument bekommt:

$$q(f_1(t_{j11}, .., t_{jk1}), ..., f_n(t_{j1n}, ..., t_{jkn}))$$

Damit erhält man für zwei Klauseln zwei solche q-Literale, deren Terme generalisiert werden. Das Ergebnis hat dann die Form:

$$q(f_1(u_{11}, .., u_{k1}), ..., f_n(u_{1n}, ..., u_{kn}))$$

Dann ist

$$\{(\pm)p_l(u_{11}, .., u_{k1}), ..., (\pm)p_n(u_{1n}, ..., u_{kn})\}$$

die speziellste Generalisierung der zwei Klauseln.

Wir haben gesehen, daß die Listen groß werden. Die Reduktion entfernt überflüssige Literale aus Klauseln, die nämlich verhindern würden, daß äquivalente Klauseln auch nichts anderes sind als Varianten voneinander. Alle solche L aus einer Klausel E werden entfernt, für die es eine Substitution gibt, so daß $E \setminus \{L\} \supseteq E\sigma$. Auch ohne L umfaßt E immer noch $E\sigma$. Da aber alle möglichen Teilmengen der Klausel gebildet werden müssen, ist diese Reduktion exponentiell.

Diese Formalisierung hat zwei Schwächen: sie ist ineffizient und sie berücksichtigt kein Hintergrundwissen. Um auch eine Theorie einzubeziehen, hat Plotkin ein erweitertes Verfahren vorgeschlagen [Plotkin, 1971]. Danach ist eine Klausel C1 eine Generalisierung bezüglich einer Theorie T und einer anderen Klausel C2, wenn es eine Substitution θ gibt, so daß

$$T \models \forall(C1\theta \rightarrow C2)$$

θ ist zum Beispiel eine Substitution, die für alle Variablen neue Konstante einführt. Leider ist das Problem, eine speziellste Generalisierung zu finden, die mit einer gegebenen Theorie und Beispielen konsistent ist, im allgemeinen Fall nicht lösbar. Das liegt an der Konsistenzforderung.

Plotkins erweitertes Verfahren produziert einige kontraintuitive Generalisierungen. Nehmen wir zum Beispiel die folgenden Klauseln als Hintergrundwissen T an:

$$katze(X) \rightarrow haustier(X)$$
$$klein(X) \ \& \ flauschig(X) \ \& \ haustier(X) \rightarrow kuscheltier(X)$$

Nach [Plotkin, 1971] wäre dann

$$katze(X) \rightarrow klein(X) \geq flauschig(X) \;\&\; katze(X) \rightarrow kuscheltier(X)$$

Das entspricht nicht unserer Intuition, nach der die beiden Klauseln von ganz verschiedenen Begriffen handeln (nämlich *klein* und *kuscheltier*). Deshalb hat Wray Buntine eine generalisierte θ-Subsumption eingeführt, mit der die Induktion einer Klausel aus einer anderen Klausel bezüglich einer Theorie beschrieben werden kann [Buntine, 1988].

7.1.2 Generalisierte θ-Subsumption

Wir machen uns wieder die Generalisierungsbeziehung erst semantisch klar an den Modellen einer Theorie und Klauseln und dann syntaktisch an der Form der Klauseln. Für den Bezug zwischen einer Formel und den Objekten, für die die Formel wahr ist, nehmen wir wieder unser Abgleichsprädikat *cover*. Wir sehen also in der Interpretation nach, welche Objekte von einer Klausel abgedeckt werden.[9] Wir nehmen das Hintergrundwissen insofern hinzu, als wir nur in Modellen der Theorie nachsehen.

> $C1 \geq_T C2$ für alle Interpretationen I, so daß T in I gilt, und für alle Atome A gilt: wenn $covers(C2, A)$ dann auch $covers(C1, A)$.

C1 ist genereller als C2 bezüglich einer Theorie T, wenn in jeder Interpretation I, die T wahr macht, für alle Atome A gilt, daß wann immer C2 auf A zutrifft, dann trifft auch C1 auf A zu. Die ist die Bedeutung der Generalisierung mit Hintergrundwissen. Damit können wir aber nicht arbeiten. Um eine Generalisierung zu konstruieren, brauchen wir eine Definition, die mit der Form von Klauseln auskommt.

Bei Hornklauseln nennen wir das positive Literal den Klauselkopf, die negativen Literale den Klauselkörper. Wir beziehen uns auf den Klauselkopf der Klausel C1 mit $C1_{kopf}$, entsprechend schreiben wir $C1_{körper}$ für den Klauselkörper von C1. Wenn die Substitution derart ist, daß für alle Variablen des Klauselkopfes neue Konstante eingeführt werden, so heißt sie θ. Wir belassen σ als Bezeichner für (irgend)eine Substitution von Termen. Dann definiert Buntine die Generalisierung folgendermaßen [Buntine, 1988]:

> $C1 \geq_T C2$ gdw.
> $\exists\sigma$, so daß $C1_{kopf}\sigma = C2_{kopf}$ und
> T, $C2_{körper}\theta \models \exists(C1_{körper}\sigma\theta)$

Wir müssen also die generellere Klausel durch Substitutionen erst einschränken, damit sie aus der spezielleren folgt. Aus einer Theorie, aus der wir induzieren, daß eine Eigenschaft z.B. für alle Tiere gilt, folgt, daß die Eigenschaft für *rex* gilt. Hier sieht man wieder die Beziehung zwischen Induktion und logischer Folgerung: was

[9] Praktischerweise nimmt man eine Herbrand-Interpretation, die für C1, C2 ,T konstruiert ist.

durch Induktion für alle gesagt wird, wird durch Deduktion für ein bestimmtes Atom gesagt.

Diese Definition können wir für alle Klauseln einer Menge von Klauseln anwenden, d.h. um Mengen von Klauseln zu vergleichen, beschränken wir uns auf den Vergleich aller einzelnen Klauseln der Mengen.

Die Definition operationalisiert Buntine, indem er für jede Klausel C_i der generelleren Klauselmenge zeigt, daß sie zurückgeführt werden kann auf eine Klausel der spezielleren Klauselmenge, indem

- Variable aus C_i in Konstante oder andere Terme überführt werden,
- Atome dem Klauselkörper von C_i hinzugefügt werden, oder
- der Klauselkörper von C_i im Hinblick auf die Theorie teilweise ausgewertet wird , d.h. ein Atom aus C_i wird mit einer Klausel der Theorie resolviert.

Dieses Verfahren ist entscheidbar, wenn die Theorie keine Funktionen enthält. Da mit θ die Variablen gleicher Prädikatssymbole im Klauselkopf der generelleren und der spezielleren Klausel unifiziert werden, kommen die kontraintuitiven Effekte der Definition von Plotkin nicht mehr vor.

Wie sieht es bei den folgenden Klauseln aus, ist C1 $\geq$ C2 bezüglich T?

$$\begin{array}{lll} T & sei: & tier(X) \ \& \ im\_haus(X) \rightarrow haustier(X) \\ C1 & sei: & flauschig(Y) \ \& \ haustier(Y) \rightarrow kuscheltier(Y) \\ C2 & sei: & flauschig(Z) \ \& \ tier(Z) \ \& \ im\_haus(Z) \rightarrow kuscheltier(Z) \end{array}$$

Wir erhalten als T, $C2_{k\ddot{o}rper}\theta$ mit $\theta : \{Z/b\}$:
$tier(X) \ \& \ im\_haus(X) \rightarrow haustier(X)$
$flauschig(b) \ \& \ tier(b) \ \& \ im\_haus(b)$

und folgern mit $\sigma 1 : \{X/b\}$ und der Schnittregel
$flauschig(b) \ \& \ haustier(b)$
Das ist $C1_{k\ddot{o}rper}\sigma\theta$, wenn wir σ:{} wählen, $\theta : \{Y/b\}$.

Es gilt also C1 $\geq$ C2 bezüglich T.

Das Verfahren von [Buntine, 1988] zählt also nicht nur die Literale durch, sondern berücksichtigt, daß einige Literale (hier: $tier(X)$ und $im\_haus(X)$) durch die Theorie verbunden sind mit einem anderen Literal (hier: $haustier(X)$).

Die Faustregel ist jetzt mit den Substitutionen σ und θ und der Einbeziehung der Theorie so formalisiert, daß sie Ergebnisse liefert, die unserer Intuition entsprechen. Ihre Operationalisierung verwendet die Resolution, um alle Literale herauszuschneiden, die sowieso durch die Theorie gegeben sind. Die Substitutionen sorgen dafür, daß keine unzusammenhängenden Objekte einbezogen werden. Dann können wir einfach abzählen, welche Klausel mehr Literale enthält.

7.1.3 RDT – Generalisierung über Regelschemata

Ein Lernverfahren, das sowohl die Generalisierungsbeziehung zwischen Klauseln als auch eine weitere Vorstrukturierung des Hypothesenraums durch eine Generalisierungsbeziehung zwischen Regelschemata verwendet, ist RDT citeKietz/Wrobel/92. Das Lernverfahren ist Teil des Systems MOBAL. Es soll hier nicht in allen Einzelheiten beschrieben werden. Interessant in diesem Zusammenhang ist nur, wie die θ-Subsumption auch für Mengen syntaktisch gleicher Regeln eingesetzt werden kann.

Mengen syntaktisch gleicher Regeln kann man durch Regelschemata ausdrücken, in denen Prädikatsvariable anstelle von Prädikatssymbolen stehen. Alle Instanzen eines Regelschemas sind von der gleichen Form und unterscheiden sich nur durch die eingesetzten Prädikatssymbole. Gemäß der Faustregel für die θ-Subsumption ist ein längeres Regelschema auch ein spezielleres Regelschema, solange man nicht in den Prämissen dasselbe Prädikatssymbol für verschiedene Prädikatsvariablen einsetzt. Eine Substitution Σ substituiert Prädikatsvariablen, ohne verschiedene Prädikatssymbole zu unifizieren. Man kann also die Suche nach der generellsten Regel, die alle positiven Beispiele abdeckt und kein negatives, dadurch strukturieren, daß man zunächst alle möglichen Instanzen für das kürzeste Regelschema überprüft und dann – solange noch negative Beispiele abgedeckt werden – jeweils die nächst längeren Regelschemata mit allen möglichen Instanzen testet. Man braucht für eine Instanz keine weitere Spezialisierung zu versuchen, wenn schon für diese Instanz zu wenig Beispiele in den Fakten zu finden sind. Dadurch, daß die Spezialisierung abgebrochen wird, sobald eine generellste diskriminierende Regel (*most general discrimination* – MGD) gefunden ist, werden keine Regeln mit redundanten Literalen gelernt.

Damit keine von der Konklusion ganz unabhängigen Argumente in eine Hypothese einbezogen werden, wird die Instanziierung zusätzlich beschränkt. Die Prämissen werden danach geordnet, wie direkt ihre Argumente mit dem (oder den) Argument(en) der Konklusion verbunden sind. Dabei ist das Argument der Konklusion mit sich selbst am direktesten verbunden. Dasselbe Argument in einer mehrstelligen Prämisse kann ein anderes Argument mit der Konklusion verbinden. Dieses andere Argument kann dann in einer mehrstelligen Prämisse ein weiteres Argument einbinden, und so fort. Prädikate können nur so in ein Regelschema eingesetzt werden, daß ihre Argumente mit der Konklusion verbunden sind.

Ein Regelschema RS ist genereller als ein anderes RS', wenn es ein Σ gibt, so daß RS' $\supseteq$ RS $\Sigma\sigma$. Die Regelschemata werden MOBAL vom Benutzer eingegeben und dann vom System auf ihre Generalisierungsbeziehung hin geordnet. Ein Beispiel für eine solche Anordnung von Regelschemata zeigt das Bild 9.

Es kann verschiedene $\Sigma\sigma$ für eine Spezialisierung geben. Nehmen wir beispielsweise an, das Regelschema $\rightarrow Q(X)$ sei als $essbar(X)$ instanziiert. Dann seien als

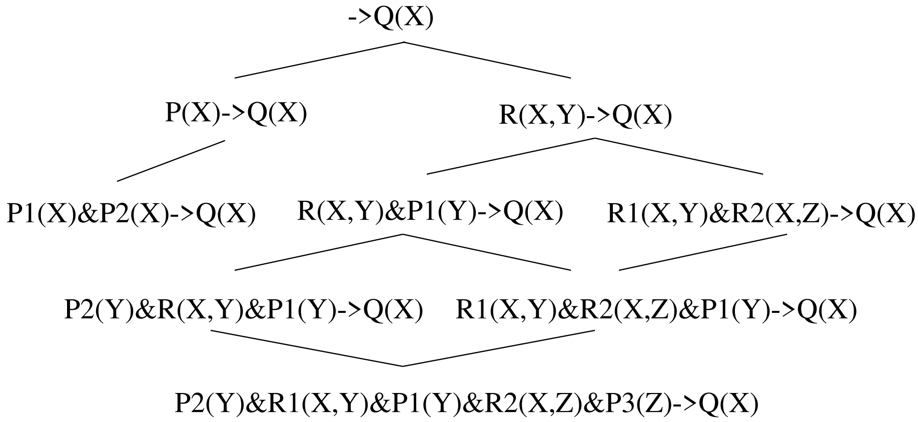

Abbildung 9: Hierarchie von Regelschemata

nächstes die beiden Spezialisierungen

$pflanze(X) \rightarrow essbar(X)$ und
$frisst(X, Y) \rightarrow essbar(X)$

getestet worden. Es gab aber nicht-eßbare Pflanzen und nicht-eßbare Objekte, die etwas anderes fressen. Es soll also weiter spezialisiert werden. Für die erste Hypothese $pflanze(X) \rightarrow essbar(X)$ kann das speziellere Regelschema

$$P1(X)\ \&\ P2(X) \rightarrow Q(X)$$

instanziiert werden. Das P des generelleren Schemas kann auf $P1$ oder $P2$ des ersten spezielleren Schemas abgebildet werden. In diesem Falle ändert das nicht viel. Es kann aber vorkommen, daß je nach Σ zusätzliche Prädikate über verschiedenen Objektbereichen gesucht werden. So zum Beispiel, wenn $frisst(X, Y) \rightarrow essbar(X)$ schon weiter spezialisiert wurde zu

$$frisst(X, Y)\ \&\ nachkomme(X, Z)\ \&\ pflanze(Y) \rightarrow essbar(X)$$

und nun weiter spezialisiert werden soll. Dann müssen für das nächst speziellere Regelschema den Prädikatsvariablen $P1, P2, P3, R1, R2$ die Prädikate $frisst$, $nachkomme$, $pflanze$, $essbar$ zugeordnet und die entsprechend noch nicht instanziierten Prädikatsvariablen mit zusätzlichen Prädikaten belegt werden. In das Regelschema

$$P2(Y)\ \&\ R1(X, Y)\ \&\ P1(Y)\ \&\ R2(X, Z)\ \&\ P3(Z) \rightarrow Q(X)$$

können die bereits ausgewählten Prädikate auf zwei verschiedene Weisen auf die Prädikatssymbole verteilt werden.

$$\Sigma 1 : \{R1/frisst, R2/nachkomme, P1/pflanze\}.$$

Dann müssen zwei neue Prädikate, eine über Pflanzen und eins über die Nachkommen gesucht werden, etwa $ungiftig$ und ei.

$$\begin{array}{l} ungiftig(Y)\ \&\ frisst(X,Y)\ \& \\ pflanze(Y)\ \&\ nachkomme(X,Z)\ \&\ ei(Z)\quad \rightarrow essbar(X) \end{array}$$

Eine andere Substitution ist

$$\Sigma 2 : \{R1/nachkomme, R2/frisst, P3/pflanze\}.$$

Dann müssen zwei neue Prädikate über die Nachkommen gesucht werden, etwa $gross$ und ei.

$$\begin{array}{l} ei(Y)\ \&\ nachkomme(X,Y)\ \& \\ gross(Y)\ \&\ frisst(X,Z)\ \&\ pflanze(X)\quad \rightarrow essbar(X) \end{array}$$

Im ersten Fall wurde nach einer weiteren Eigenschaft für Pflanzen ($ungiftig$), im zweiten nach einer weiteren Eigenschaft für Nachkommen ($gross$) gesucht. Beide Hypothesen sind gleich speziell. Beide Hypothesen werden getestet, ob sie einem vom Benutzer gegebenen Bewertungskriterium genügen. Beim Testen werden die Argumentvariablen durch konstanten Terme substituiert, so daß sie positiven oder negierten Fakten entsprechen.

Der Suchraum von RDT ist nicht nur durch die Generalisierungsbeziehung zwischen Regelschemata strukturiert, sondern auch noch durch andere Verfahren beschränkt, auf die hier aber nicht eingegangen werden soll. Hier sollte nur dargestellt werden, wie nicht nur die Generalisierungsbeziehung zwischen Regeln, sondern auch die zwischen Regelmengen formalisiert werden kann.

7.2 Induktion als inverse Resolution

Ein naheliegender Gedanke ist es, die Induktion als umgekehrte Deduktion aufzufassen. Die Resolution hat sich als operationales Verfahren für die Deduktion bewährt. Wäre eine inverse Resolution dann vielleicht ein operationales Verfahren für die Induktion? Diese Frage untersuchten zeitgleich Rüdiger Wirth für die Vervollständigung einer Menge von Syntaxregeln für ein natürlichsprachliches System [Wirth, 1989] und Stephen Muggleton und Wray Buntine [Muggleton und Buntine, 1988]. In der Folge haben vor allem Celine Rouveirol und Jean-Francois Puget daran weitergearbeitet [Rouveirol und Puget, 1990]. Im folgenden wird das Verfahren in seinen Grundzügen dargestellt und auf seine Schwierigkeiten hingewiesen. Es ergibt sich dann, daß die inverse Resolution nicht so effizient ist wie andere logik-basierte Verfahren, es sei denn man würde das Verfahren stark verändern.
Eine kurze Rekapitulation der Resolution:

> Seien B und D zwei Klauseln, $\neg$ R1 ein negatives Literal aus B und R2 ein positives Literal aus D;
> Wenn es einen generellsten Unifikator σ gibt mit $R1\sigma = R2\sigma$,
> dann sind R1 und R2 Resolutionsliterale und
> $C = (B - \neg R1)\sigma \cup (D - R2)\sigma$ der Resolvent.

Ein Beispiel:

B D

$B : \neg f(U,U,U), h(U,V,W), a$
$\neg R1 : \neg f(U,U,U)$
$\sigma 1 : \{U/S\}$

$D : f(X,Y,Z), g(X,Y,Z), a$
$R2 : f(X,Y,Z)$
$\sigma 2 : \{X/S, Y/S, Z/S\}$

C

$C : h(S,V,W), a, g(S,S,S)$
$\sigma : \{U/S, X/S, Y/S, Z/S\}$

Abbildung 10: ein Resolutionsschritt

Mit diesen Ausdrücken können wir die **inverse Resolution** als die folgende Lernaufgabe angeben:

Gegeben:
Ein Resolutionsliteral B und ein Resolvent C

Ziel:
konstruiertes Resolutionsliteral D zu B und C.

Dabei gibt es zwei Probleme: die Umkehrung der Substitution und die Entscheidung, welcher nicht resolvierte Rest auf beiden Seiten gleichermaßen vorhanden ist. So ist im Beispiel nicht festzustellen, wenn wir nur B und C kennen, ob h und a auch in D vorkommen, oder nur eines von beiden. Wenn es keinen nicht-resolvierten Rest gibt, so ist D:

$$(C - B\sigma 1 \cup \neg R1\sigma 1)\sigma^{-1}$$

Dabei ist σ^{-1} die inverse Substitution. Wirth nimmt als Umkehrung der Substitution einfach neue Variablen für die verschiedenen Argumentstellen [Wirth, 1989].

Im allgemeinen Fall, wenn es einen nicht-resolvierten Rest gibt, so muß die Potenzmenge dieses Restes, $pot(B - \neg R1)$, gebildet werden. Damit gibt es im allgemeinen Fall bei n Literalen in B 2^{n-1} Lösungen für D. In unserem Beispiel sind die möglichen Lösungen:

$g(X1, X2, X3), f(X4, X5, X6)$ oder
$g(X1, X2, X3), f(X4, X5, X6), a$ oder
$g(X1, X2, X3), f(X4, X5, X6), h(S, V, W)$ oder
$g(X1, X2, X3), f(X4, X5, X6), h(S, V, W), a$

Die inverse Resolution ist also ein exponentielles Verfahren. Außerdem erfordert es die Vorgabe des Resolventen, was in der Praxis selten möglich ist. Einen Überblick über die verschiedenen Generalisierungsoperatoren bietet [Jung, 1993].

8 Lernen als nicht-monotoner Schluß

Wenn Lernen (bzw. der induktive Schluß) als inverse Resoultion nicht günstig operationalisierbar ist, so vielleicht, indem man diesen Schluß unter die nicht-monotonen Schließweisen einordnet? Monoton ist ein Schluß, wenn durch Hinzufügen von neuen Aussagen keine bisherigen Folgerungen ungültig werden. Monotonie ist also:

> Wenn $T \models X$ so auch $T \cup N \models X$, wobei T ein Theorie, N eine neue Aussage oder eine Menge neuer Aussagen, X eine Aussage oder eine Menge von Aussagen ist.

Nicht-monoton ist ein Schluß, für den diese Eigenschaft nicht garantiert ist.

Durch das Hinzufügen von Lernergebnissen wird zwar nichts falsch, was vorher wahr war. Aber es werden mögliche Modelle des gegebenen Wissens ausgeschlossen. Insofern kann man alle Lernverfahren als nicht-monotone Verfahren ansehen. Dieser Sichtweise ging Helft nach [Helft, 1989]. Seine Lernaufgabe:

Gegeben:
D: Wissen eines Sachbereichs und Beobachtungen

Ziel:
Eine Generalisierung G, die aus D induziert ist

Wie sieht nun die induktive Ableitung aus? Helft verwendet dafür eine zweistufige Bewertung, die Formeln anhand ihrer Folgerbarkeit aus minimalen Modellen von D beurteilt, und dann Bewertungen von Formeln für alle minimalen Modelle erstellt.

> Die generellste Generalisierung für D sind dann alle Formeln r,
> deren Bewertung 1 ist,
> die nicht schon (deduktiv) aus D folgen und
> für die es keine generellere Generalisierung gibt.

Ein minimales Modell von D enthält genau die Interpretation aller Aussagen aus D und nicht mehr.[10] Helft ergänzt für alle konstanten Terme aus D negierte Aussagen, wenn es über sie keine positiven Aussagen in D gibt. Dies entspricht einer *closed world assumption*, weil keine weiteren Aussagen als nur die durch D gegebenen in dem Modell gültig sind. Würde Helft sich auf Hornformeln beschränken, so gäbe es überhaupt nur ein minimales Modell. Er nimmt aber g-Klauseln (*groundable clauses*). Das sind Klauseln, bei denen keine weitere Einschränkung gemacht wird, als daß zu jeder Variablen aus einem positiven Literal (also der Konklusion) auch dieselbe Variable in einem negativen Literal (also in der Prämisse) vorkommt und keine Funktionen als Terme autreten. Man kann also eine Disjunktion in der Konklusion haben und man braucht keine Funktionen zu berücksichtigen. Außerdem

[10] Man bedenke aber, daß bei Helft nicht nur Beobachtungen, sondern auch das Hintergrundwissen zu D gehören!

muß eine Klausel injektiv über Grundformeln sein. Das heißt, für jedes Paar von Variablen X, Y einer Klausel gibt es eine Substitution, so daß $X\sigma \neq Y\sigma$ und $X\sigma$, $Y\sigma$ sind Grundinstanzen.

Eine Formel r erhält bezüglich eines Modells M

die Bewertung $Val(r, M) = 1$,
wenn M ein Modell ist für r
und es für die Prämisse von r Grundinstanzen in M gibt
und die Prämisse injektiv über dem Modell ist.

Sonst erhält sie die Bewertung $Val(r, M) = 0$.

Im zweiten Schritt erhält eine Formel r bezüglich aller minimalen Modelle

den Wert $Val(r, D) = 1$, wenn sie für alle minimalen Modelle den Wert 1 hatte,
den Wert $Val(r, D) = 0$, wenn sie für alle minimalen Modelle den Wert 0 hatte,
den Wert $Val(r, D) = 0.5$, wenn sie für mindestens ein minimales Modell, aber nicht alle, den Wert 1 hatte.

Damit läßt sich dann die Generalisierung G für das Sachbereichswissen D so angeben:

$$G(D) = \{r \mid Val(r, D) = 1 \,\&\, \neg(D \models r) \,\&\, \text{wenn } r' \in G(D) \,\&\, r' \models r \text{ dann } r \models r'\}$$

Alle solche Formeln r sind Generalisierungen von D, die in den Modellen gültig sind, aber nicht schon logisch folgern. Daß sie auch die allgemeinsten Generalisierungen sind, legt er durch die dritte Bedingung fest: jede andere Generalisierung r' kann nur äquivalent mit r sein, wenn es eine Folgerungsbeziehung zwischen r' und r gibt. Helft induziert also generellste Formeln (MGDs) und nicht speziellste. Daß sie dennoch nicht überallgemein sind, erreicht er durch die *closed world assumption*.

Ein Beispiel soll dies verdeutlichen. Nehmen wir als Sachbereichswissen D:

$fliegt(tweety)$,
$vogel(tweety)$,
$vogel(polly)$,
$\forall X \mid vogel(X) \rightarrow federn(X)$

Dann ist das minimale Modell mit der *closed world assumption*:

$fliegt(tweety)$,
$vogel(tweety)$,
$federn(tweety)$,
$vogel(polly)$,
$federn(polly)$,
$\neg fliegt(polly)$

Für die folgenden beiden Formeln gelten die für $G(D)$ angegebenen Bedingungen, d.h. sie sind gültig in dem Modell, werden aber nicht schon logisch gefolgert und sind maximal generell.

$$\begin{array}{ll} G(D): & \forall X \mid fliegt(X) \rightarrow vogel(X) \\ & \forall X \mid fliegt(X) \rightarrow federn(X) \end{array}$$

Das sind recht genau diejenigen Formeln, die der Intuition entsprechen. Insbesondere wurden durch die *closed world assumption* Fehlschlüsse, wie etwa *alles, was Federn hat, fliegt*, vermieden. Dadurch, daß über die Semantik, die den gegebenen Aussagen des Sachbereichs zugrunde liegen (eben das Modell), die Generalisierungen gefunden wurden, erreicht dieses Verfahren meist einleuchtende induzierte Formeln.

9 Theorie des Lernbaren

Die Theorie des maschinellen Lernens behandelt die Frage: was ist überhaupt lernbar und unter welchen Umständen? Je nach dem Lernproblem und den dabei gemachten Annahmen können drei Ansätze zur Theorie unterschieden werden:

- logik-orientiertes Lernen
- Identifikation im Grenzwert
- wahrscheinlich annähernd korrektes Lernen.

Zum ersten Bereich wurden im vorigen Beispiel schon einige Lernbarkeitsergebnisse angeführt. Deshalb wird nur ein kurzer Überblick gegeben. Die beiden anderen Bereiche werden oft zusammengefaßt unter verschiedenen Titeln wie *computational learning theory* oder algorithmisches Lernen. Die Theorie maschinellen Lernens ist zu umfangreich, um sie hier ausführlich darzustellen. Stattdessen werden Fragestellungen und schlaglichtartig einige Ergebnisse vorgestellt.

9.1 Logik-orientiertes Lernen

Das Lernproblem des logik-orientierten Lernens wie es als induktiver Schluß in Abschnitt 3.2 dargestellt wurde, ist in seiner Komplexität abhängig von dem Repräsentationsformalismus, in dem die Beispiele gegeben werden, also der Beschreibungssprache LO, und dem Repräsentationsformalismus, in dem die Hypothesen aufgestellt werden, also der Hypothesensprache LH.

Für die uneingeschränkte Prädikatenlogik als LO und LH gilt:

Wenn es für eine Menge von Klauseln eine Selektion gibt,[11] so gibt es auch eine speziellste Generalisierung für diese Menge von Klauseln [Plotkin, 1970]. Allerdings

[11] Eine Selektion sucht für eine Menge von Klauseln die Literale mit demselben Prädikatensymbolen und demselben Vorzeichen (negiert oder nicht negiert) aus verschiedenen Klauseln heraus.

ist nicht entscheidbar, ob die speziellste mit einer Theorie und Beispielen konsistente Generalisierung gefunden wird [Plotkin, 1971]. Das Lernproblem "erbt" die Probleme der logischen Folgerung.

Einschränkungen werden jetzt für die Beschreibungssprache, die Theorie und die Hypothesensprache formuliert, um das Lernproblem einfacher zu machen. So werden etwa Funktionen nicht als Terme zugelassen (**funktionsfreie Terme**) und die Beispiele dürfen keine Variablen enthalten (nur Grundbeispiele).

Die Theorie kann, falls sie keine Variablen enthält,[12] die in die Beispiele hineingerechnet werden, so daß wir nur noch über das Lernen aus Beispielen zu sprechen brauchen und die Theorie außer acht lassen können. Diese Einschränkung des Lernproblems heißt **Saturierung** [Rouveirol, 1991]. Die Theorie wird mit den Beispielen E so verknüpft, daß alles, was für die Beispiele von der Theorie wichtig ist, Teil der Beispiele wird. Im allgemeinen Fall kann aus der Vereinigung aller Klauseln der Theorie eine Klausel T gebildet werden. Die neuen Beispiele E_{new} sind dann Klauseln $e \leftarrow T$, wobei $e \in E$. Die Beispiele werden mit der Theorie *saturiert*. Üblicherweise nehmen wir nicht das gesamte Hintergrundwissen zu jedem Beispiel hinzu, sondern nur solche Klauseln, deren Argumente mit dem Beispiel unifiziert werden können oder mit unifizierbaren Argumenten verbunden sind. Die Menge der konstanten Terme aus dem Beispielfakt – also dem Klauselkopf von E_{new} – ist eine Teilmenge der konstanten Terme aus dem hinzugenommenen Teil der Theorie – also dem Klauselkörper von E_{new}. Damit sind die Beispiele Klauseln und zwischen funktionsfreien, nicht rekursiven Hornklauseln ist die θ-Subsumption eine korrekte und vollständige Ableitung.

Wenn die Theorie noch weiter beschränkt wird auf eine feste und höchste Stelligkeit der Prädikate und die Klauseln generativ sind, so können alle Ableitungen mit der Theorie in einer Zeit polynomiell zur Größe der Theorie durchgeführt werden. Eine Klausel ist **generativ**, wenn jede Variable aus dem Klauselkopf auch im Klauselkörper vorkommt. Eine noch stärkere Einschränkung stellen die beschränkten Klauseln dar (*constrained clauses*)[Page und Frisch, 1992]. Eine Klausel ist **beschränkt**, wenn alle Variablen, die im Klauselkörper vorkommen, auch im Klauselkopf vorkommen. Wenn solche beschränkten Klauseln als Hypothesensprache gewählt werden, ist das Lernproblem wahrscheinlich annähernd korrekt lösbar [Page und Frisch, 1992]. Diese Aussage über die Lernbarkeit verknüpft zwei Ansätze des theoretischen maschinellen Lernens, das logik-basierte und das wahrscheinlich annähernd korrekte Lernen (zu letzerem s. Abschnitt 9.3).

Eine wichtige Einschränkung der Hypothesensprache ist die auf ij-deterministische Klauseln, weil für diese gezeigt wurde, daß sie in polynomieller Zeit zu lernen sind, wenn das Hintergrundwissen aus Grundbeispielen besteht und die Beispiele Fakten – ebenfalls ohne Variablen – sind. **ij-deterministische Klauseln** sind deterministische Klauseln, deren maximale deterministische Tiefe i und deren

[12] Für eine endliche Menge von variablenfreien Beispielen können wir immer die Theorie ebenfalls variablenfrei machen, indem alle substituierbaren konstanten Terme der Beispiele für die Variablen in der Theorie eingesetzt werden.

maximale Stelligkeit von Prädikaten j ist. Eine Klausel ist **deterministisch**, wenn jeder ihrer Terme deterministisch verbunden ist.

Wir nehmen an, wir haben Beispiele E und eine Theorie T als Hintergrundwissen. Ein Term aus dem Klauselkopf A ist mit einer Kette der Länge 0 **deterministisch verbunden**. Für den Klauselkörper nehmen wir an, daß die Literale nach ihrer Verbundenheit mit dem Klauselkopf geordnet sind: $\{\neg B_1, ..., \neg B_m, \neg B_{m+1}, ..., \neg B_n\}$. Ein Term t aus $\neg B_{m+1}$ ist genau dann durch eine deterministische Kette der Länge d+1 verbunden, wenn

- alle Terme im Klauselkopf und in $\{\neg B_1, ..., \neg B_m\}$ verbunden sind durch Ketten, die höchstens d lang sind, und
- es für jede Substitution θ, die A mit einem Beispiel und die ersten Literale mit dem Hintergrundwissen unifiziert – (A$\theta \in E^+$) und $\{\{B_1\}, ..., \{B_m\}\}\theta \subseteq T$ –, genau eine eindeutige Substitution δ gibt, so daß $B_{m+1}\theta\delta \in T$.

Die minimale Länge der deterministisch verbindenden Ketten ist die deterministische Tiefe eines Terms.

Stephen Muggleton und Cao Feng haben gezeigt, daß man ij-deterministische Klauseln in polynomieller Zeit lernen kann, wenn variablenfreie Fakten als Beispiele und variablenfreie Klauseln als Theorie gegeben sind [Muggleton und Feng, 1992]. Damit haben sie eine obere Schranke der Lernbarkeit angegeben. Die Verbindung zum wahrscheinlich annähernd korrekten Lernen haben Saso Dzeroski, Stephen Muggleton und Stuart Russell hergestellt, indem sie für einfache Verteilungen der positiven und negativen Beispiele beweisen, daß k funktionsfreie, nicht rekursive ij-deterministische Klauseln wahrscheinlich annähernd korrekt lernbar sind [Dzeroski et al., 1992]. Im Grunde ist die Hypothesensprache dann gar nicht aussagekräftiger als Aussagenlogik. Aber in vielen Fällen ist die Repräsentation dann leichter verständlich, also für einen Menschen leichter lesbar.

Untere Schranken wurden jetzt erstmals von Jörg-Uwe Kietz gefunden [Kietz, 1993b]. Er zeigt, daß das Konsistenzproblem für nicht tiefenbeschränkte deterministische Hornklauseln der Stelligkeit 2 in PSPACE liegt [Kietz, 1993b]! Selbst das Lernen indeterministischer Klauseln der Tiefe 1 und der Stelligkeit 2, also das Lernen 12-indeterministischer Klauseln, kann auf das Problem der Erfüllbarkeit einer Formelmenge, SAT, zurückgeführt werden und ist damit NP-hart [Kietz, 1993b]. 12-indeterministische Hornklauseln sind nicht wahrscheinlich annähernd korrekt lernbar, es sei denn RP=NP. Auch Jörg-Uwe Kietz setzt die Ergebnisse über die Lernbarkeit des induktiven Lernproblems mit den Annahmen der Ansätze des wahrscheinlich annähernd korrekten Lernens in Bezug.

Die Ergebnisse zeigen deutlich, daß für die effiziente Lernbarkeit sehr enge Grenzen gesetzt sind. Es gibt nur die Einschränkungen der Repräsentationssprachen, die zwischen den beiden angegebenen Schranken liegen. Die meisten Arbeiten beschäftigen sich mit geeigneten Einschränkungen der Hypothesensprache. So ist zum Beispiel auch eine Termsubsumptionssprache (wie etwa KL-ONE) eine mögliche Hypothesensprache, die polynomielles Lernen erlaubt [Kietz und Morik, 1993].

Man könnte sich aber auch noch andere, zusätzliche Einschränkungen des Lernproblems vorstellen:

- *Beispiele* werden in einer günstigen Reihenfolge gegeben, wobei festgelegt werden muß, wann eine Reihenfolge günstig ist;
- *Beispiele* werden sorgfältig ausgewählt, um das Lernen zu erleichtern (im Sinne des Erziehens, das Turing vorschlug [Turing, 1987] oder des abgrenzenden Gegenbeispiels, das Winston einführte [Winston, 1987]);
- einige Entscheidungen im *Lernprozeß* werden vom Benutzer (Lehrer) übernommen;
- das *Lernverfahren* durchsucht den Hypothesenraum nicht vollständig, sondern wird durch Heuristiken beschränkt;
- das *Lernergebnis* ist nicht die speziellste oder generellste Generalisierung, sondern irgendeine Generalisierung.

9.2 Identifikation im Grenzwert

Den Arbeiten zur **Identifikation im Grenzwert** (identification in the limit) liegt folgende Vorstellung zugrunde. Es geht beim Lernen um das Ermitteln einer Theorie oder Funktion oder Sprache anhand einer Folge von Eingaben(wahre und falsche Fakten, Werte aus dem Definitionsbereich und zugehöriger Wert aus dem Wertebereich einer Funktion, Wörter einer Sprache). Nach jeder solchen Eingabe gibt das lernende System ein Lernergebnis aus. Dieser Prozeß geht ewig so weiter. Ein Lernergebnis **erklärt** ein Modell einer Theorie oder eine Funktion oder eine Sprache, wenn das lernende System nach diesem Lernergebnis auf alle folgenden Eingaben nur noch mit syntaktischen Varianten des Lernergebnisses reagiert. In gewisser Weise entspricht dieser Begriff der Erklärung dem der Beschreibungsadäquatheit in der Linguistik. Das Lernergebnis ist sozusagen beschreibungsadäquat, weil es auch neue Eingaben richtig beantwortet. Auch zu der Beobachtungsadäquatheit in der Linguistik gibt es eine Entsprechung in der Lerntheorie: Ein Lernergebnis **beschreibt** ein Modell oder eine Funktion oder eine Sprache, wenn alle folgenden Reaktionen des Systems ebenfalls richtig sind. Hier muß dem Lernergebnis nicht die richtige Theorie, die richtige Funktionsdefinition oder die richtige Grammatik zugrunde liegen, aber es muß zur jeweils richtigen Reaktion auf eine Eingabe führen. Insofern kann das Lernergebnis dann – in einer Analogie – beobachtungsadäquat im linguistischen Sinne genannt werden.

Der einfachste Lernalgorithmus ist der Aufzählungsalgorithmus, der alle Theorien, Funktionen, Sprachen aufzählt. Er rät einfach ein Ergebnis und, wenn sich dieses Ergebnis bei der nächsten Eingabe als falsch herausstellt, nimmt er das nächste. Nehmen wir diesen einfachsten Algorithmus als Grundlage, wir können uns aber auch jeden anderen denken. Dann **identifiziert** der Algorithmus das richtige Ergebnis (die Theorie, die Funktion, die Sprache) im Grenzwert, wenn, nachdem einmal (im Grenzwert) das richtige Ergebnis gefunden wurde, nie wieder ein anderes gewählt wird. Die Bedingung fordert, daß irgendwann das Richtige gefunden wird. Dies ist dann der Grenzwert. Ab diesem Zeitpunkt verändert sich das Lernergebis nicht mehr. Die Bedingung fordert nicht, daß das Lernverfahren oder irgendjemand sonst bemerkt, daß jetzt das Richtige gefunden ist. Der Grenzwert ist also unbekannt. Einen guten Überblick zu diesem Szenario und den darin erforschten Bereichen geben Angluin und Smith in der "Encyclopedia of Artificial Intelligence" oder auch [Angluin und Smith, 1983].

Ein Beispiel soll deutlich machen, warum es gar nicht möglich ist, zu wissen, wann das richtige Ergebnis erreicht wurde. Man könnte ja meinen, daß, wenn das Ergebnis eines Algorithmus', von dem nachgewiesen wurde, daß er im Grenzwert identifiziert, sich längere Zeit nicht verändert, dieses dann wohl das richtige ist. Das Beispiel zeigt, daß es für jede längere Zeit eine noch längere gibt, in der sich das Ergebis als falsch herausstellen kann. Das Beispiel handelt vom Identifizieren einer Funktion. Der Definitions- und der Wertebereich sind die natürlichen Zahlen. Das, was aufgezählt wird, sind Funktionen, hier speziell: alle Polynome mit nur einer Variablen. Als Lernergebnis wird ein Polynom zur Berechnung der Funktion ausgegeben. Die Beispiele für die Funktion p werden dem System in Form von Paaren $(n, p(n))$ in aufsteigender Reihenfolge von n ($n \in \mathbb{N}$) eingegeben.

Eingabe: (0,1) Ausgabe: 1
Eingabe: (1,1) Ausgabe: 1
Eingabe: (2,1) Ausgabe: 1
Eingabe: (3,1) Ausgabe: 1
Eingabe: (4,1) Ausgabe: 1

Nun könnte man allmählich meinen, die Wahrheit identifiziert zu haben: es handelt sich um das konstante Polynom 1! Nehmen wir also an, daß nach fünfmaliger Wiederholung des Ergebnisses das richtige gefunden ist. Aber dann kommt als nächstes:

Eingabe: (5, 121)

Es könnte sich etwa um die folgende Funktion handeln:

$$1 + x(x-1)(x-2)(x-3)(x-4)$$

Bei dieser Funktion ist für $x \in \{0, 1, 2, 3, 4\}$ jeweils ein Faktor gleich 0. Für jede beliebige Schwelle können wir so eine Funktion konstruieren, bei der im Schritt nach der Schwelle das so lange konstante Ergebnis nicht mehr gilt. Deshalb können wir nicht fordern, daß aufgrund unvollständiger Information (die Eingaben) bestimmt werden kann, wann das richtige Ergebnis identifiziert wurde. Wir können nur fordern, daß jedenfalls ab dem Zeitpunkt, zu dem das Richtige gelernt wurde – wann immer das sei – das Richtige nicht durch etwas Falsches ersetzt wird, sondern höchstens durch etwas genauso Richtiges.

Aber auch mit dieser Einschränkung konnte keine induktive Methode gefunden werden, die alle vollständig berechenbaren Funktionen beschreibt oder gar erklärt. Schlimmer noch: es gibt keine induktive Methode, die reguläre Sprachen im Grenzwert identifiziert – obwohl doch Kinder sogar die natürlichen Sprache ihrer Umgebung lernen! Was man tun kann, ist

- die Anforderungen noch weiter abschwächen;
- das Szenario dahingehend ändern, daß mehr Informationen in das System eingegeben werden;
- Spezialverfahren entwickeln.

Um das Paradigma der Identifikation im Grenzwert etwas anschaulicher zu machen, sei hier die Arbeit von Ehud Shapiro dargestellt, das Model Inference System (MIS) [Shapiro, 1981], [Shapiro, 1983].

9.2.1 Model Inference System MIS

Die Lernaufgabe, die das MIS löst, besteht darin, aus Fakten, die in einem logischen Modell M gelten, ein Axiomensystem T in einer dazu passenden Signatur L zu inferieren. Dabei gibt ein Orakel (z.B. der Benutzer) an, ob ein Fakt wahr ist, also im Modell gilt, oder nicht.

Gegeben:

eine Beschreibungssprache LO und eine Hypothesensprache LH,
die beide in der Signatur L sind;
ein Orakel, das zu jedem Fakt den Wahrheitswert liefert
gemäß seiner Gültigkeit in M;
ein vollständiges deduktives Ableitungsverfahren;

Ziel:

eine Theorie, dargestellt in LH, aus der alle wahren Fakten (also alle Fakten aus LO, die vom Orakel als wahr bewertet wurden) abgeleitet werden können, und keine falschen.

Das Ziel ist also eine bezüglich der Beschreibungssprache vollständige Axiomatisierung von M. Die Situation veranschaulicht Bild 11.

Dabei sollen LO und LH angemessen (*admissible*) sein, d.h. zusammen passen. Angemessen sind zwei Sprachen einer Signatur, wenn für jedes Modell der Signatur und jede Theorie, ausgedrückt in der Hypothesensprache LH, daraus, daß die wahren Fakten, ausgedrückt in LO, aus der Theorie ableitbar sind, folgt, daß die Theorie gültig ist in M. Dies ist die Grundannahme, mit der eine solche Lernaufgabe überhaupt erst möglich wird. Wenn sie nämlich nicht gelten würde, dann hätten wir die paradoxe Situation, daß eine Theorie genau alle wahren Fakten ableitet und

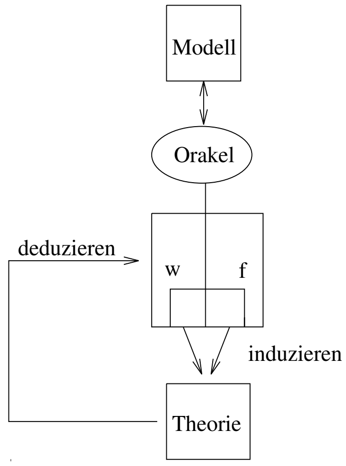

Abbildung 11: Lernsituation von MIS

doch falsch ist. Diese Situation käme insbesondere dann vor, wenn wir einen für den Wahrheitswert wichtigen Unterschied in der Beschreibungssprache nicht ausdrücken können. Shapiro beruft sich auf die erkenntnistheoretische Forderung, daß Theorien durch Fakten falsifizierbar sein sollen. Zu falschen Theorien soll es auch Fakten geben, die vom Orakel als falsch bewertet werden, so daß die Theorie abgelehnt werden kann.

Eine weitere Beschränkung des Problems besteht in der Anzahl der Ableitungsschritte, die für die Ableitungen zugelassen wird. Dabei wird nicht einfach eine konstante Beschränkung gewählt, sondern eine Funktion h, die für das i-te Fakt die Zahl der Ableitungsschritte $h(i)$ festlegt. Die Funktion h sei vollständig berechenbar und rekursiv. Wir können aufgrund dieser Beschränkung nur solche Modelle axiomatisieren, bei denen die Theorie für jedes Fakt a_i nur $h(i)$ Ableitungsschritte benötigt, um es aus der Theorie abzuleiten. Ein Modell, für das es so eine Theorie gibt, heißt h-leicht (h-*easy*).

Der Algorithmus ist zunächst ein Aufzählungsalgorithmus. Zwei Mengen von Fakten, die wahren und die falschen, werden nach ihrer Eingabe gespeichert. Solange es einen falschen Fakt gibt, der aus der Theorie abgeleitet wird, oder einen wahren Fakt a_i gibt, der nicht in $h(i)$ Schritten aus der Theorie abgeleitet werden kann, wird die nächste Theorie genommen und ausgegeben. Die Theorien stehen durchnumeriert zur Verfügung. Der Algorithmus hält nie an. Die Bedingungen für den Übergang zur nächsten Theorie sind dergestalt, daß der Algorithmus die Theorie im Grenzwert identifiziert: nur wenn Falsches abgeleitet wird oder Wahres nicht abgeleitet wird, geht der Algorithmus zur nächsten Theorie über, sonst nicht.

Natürlich ist so ein Aufzählungsalgorithmus nicht sehr befriedigend, weil eigentlich Theorien nicht durchnumeriert zur Verfügung stehen. Deshalb hat Shapiro einen Algorithmus implementiert, der gezielter nach der nächsten Theorie sucht.

Dabei benutzt er die beiden Bedingungen für die Notwendigkeit, eine andere Theorie zu finden. Eine Theorie ist zu stark, wenn sie (auch) Falsches ableitet. Dann muß sie abgeschwächt werden. Das heißt, die nächste Theorie sollte ähnlich wie die vorhergehende sein, nur schwächer. Eine Theorie ist zu schwach, wenn sie Wahres nicht ableiten kann. Dann sollte sie verstärkt werden. Statt also einfach irgendeine neue Theorie zu nehmen, verändert man gezielt die einmal gewählte Theorie. Dazu braucht man zwei Verfahren, eines zur Abschwächung und eines zur Verstärkung.

Shapiros Verfahren zur Abschwächung heißt *contradiction backtracing*. Es findet die falsche Hypothese aus der zu starken Theorie, die die Ableitung eines falschen Fakts ermöglichte. Dabei wird der Ableitungsbaum, der zum Widerspruch führt zwischen falschen Fakt und abgeleitetem Fakt ausgenutzt. Zum Beispiel könnte es den folgenden Ableitungsbaum geben, der den Widerspruch zwischen T und $\neg T$ darstellt.

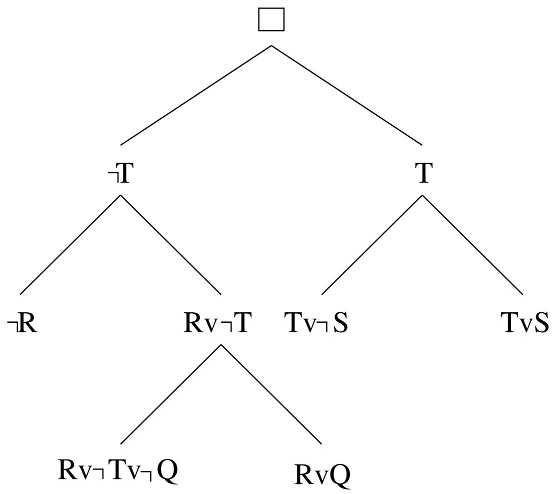

Abbildung 12: Ableitungsbaum

Der Algorithmus beginnt an der Wurzel und befragt das Orakel, ob das jeweils resolvierte Atom im Modell M wahr ist oder nicht. Zuerst wird also gefragt, ob T gilt. Dabei sind die Fragen so formuliert, daß nach einem Grundatom (aus LO) gefragt wird. Das Orakel liefert nur Wahrheitswerte für Fakten, LO enthält ja nur Fakten. Substitutionen sind bereits beim Beweis mit dem Resolutionsprinzip entstanden. Falls aber ein resolviertes Atom nicht schon ein Grundbeispiel ist (was es ja nicht zu sein braucht), so wird die Substitution passend erweitert. Wenn das Orakel sagt, daß das Atom im Modell wahr ist, so kann der Fehler nicht in der rechten Seite sein, weil dort die nicht negierten Atome stehen. Also wird dann nach links der Kante gefolgt und beim nächsten resolvierten Atom gefragt, ob es wahr ist. Im Beispiel wird dann R gefragt. Wenn das Orakel sagt, daß das resolvierte Atom falsch ist, wird nach rechts weitergegangen. Es wird auf diese Weise gezielt gefragt, bis ein Blatt erreicht ist. Dieses Blatt ist die falsche Hypothese. Sie wird aus der Theorie gelöscht. Die neue Theorie ist aus der zu starken Theorie durch Löschen einer falschen Hypothese entstanden. Sie ist damit schwächer als die vorherige Theorie. Dieses Verfahren kann als Hilfe bei der Fehlersuche von Prolog-Systemen eingesetzt werden.

Die Verstärkung einer Theorie erreicht Shapiro mit einem Verfeinerungsoperator, der schrittweise Verfeinerungen zurückgewiesener Hypothesen in die zu schwache Theorie einfügt. Der Verfeinerungsoperator findet für jede Formel P eine umfangreichere Formel Q, die von P impliziert wird. Das ist so nicht berechenbar (alles, was von P impliziert wird). Ein konkreter Verfeinerungsoperator findet daher ein Q, das mit P in einer Verfeinerungsrelation sr steht.

Für ein $P \in L$ gilt die Verfeinerungsrelation $sr(P, Q)$ gdw.

- P ist die leere Aussage und Q ist ein n-stelliges Atom $q(X_1, ..., X_n) \in L$; oder
- P ist ein Atom p mit der Variable U, dann ist Q dasselbe Atom mit einer anderen Variablen V von p und $U = V$ wird angehängt oder Q ist dasselbe Atom mit einer n-stelligen Funktion über Variablen X_i, die in p nicht vorkommen (Substitution von U durch $f(X_1, ..., X_n), f \in L$); oder
- P ist ein n-stelliges Prädikat $p(t_1, ..., t_n$, dann ist Q die Formel $p(X_1, ..., X_n) \rightarrow p(t_1, ..., t_n)$, wobei X_i in t_i vorkommt für alle $i \in \{1, ..., n\}$.

  Q ist zum Beispiel dadurch *umfangreicher*, daß es mehr Prämissen enthält als P.

Über die Verfeinerungsrelation ist eine partielle Ordnung gegeben. Ein Verfeinerungsoperator ist vollständig, wenn aus der leeren Aussage über schrittweises Verfeinern alle in L bildbaren Sätze erzeugt werden können.[13] Der Verfeinerungsoperator ist ein konstruktives Verfahren zum Hinzufügen von Hypothesen im Gegensatz zur schrittweisen Aufzählung von Theorien mit mehr Hypothesen. Der Verfeinerungsoperator geht entlang der Verfeinerungsrelation von einer Ebene von Sätzen zur nächsten Ebene von verfeinerten Sätzen. Man kann sich die Sätze also in einem gerichteten, azyklischen Graphen angeordnet vorstellen, bei dem die Kanten die Verfeinerungsrelation darstellen und die Knoten Hypothesen sind. Dabei ist der Wurzelknoten die leere Aussage. Alle Grundaussagen des Modells sind in diesem Graphen. Obendrein gilt für einen Knoten dieses Graphen, daß, wenn die entsprechende Hypothese im Modell M gültig ist, so auch die aller Nachfolgerknoten und keine der Vorgängerknoten. Die richtige Theorie für das Modell kann man sich als Linie durch den Verfeinerungsgraphen vorstellen wie in Abbildung 13 schematisch dargestellt.

MIS besteht aus dem *contradiction backtracing* und dem Verfeinerungsoperator und arbeitet wie folgt:

[13] Die Verfeinerung bei RDT (siehe 7.1.3) ist bezüglich der Hornlogik mit expliziter Negation nicht vollständig, sondern auf die vorhandenen Regelschemata eingeschränkt. Wenn man die Sprache ebenfalls auf die Schemata einschränkt, ist die Verfeinerung bei RDT vollständig.

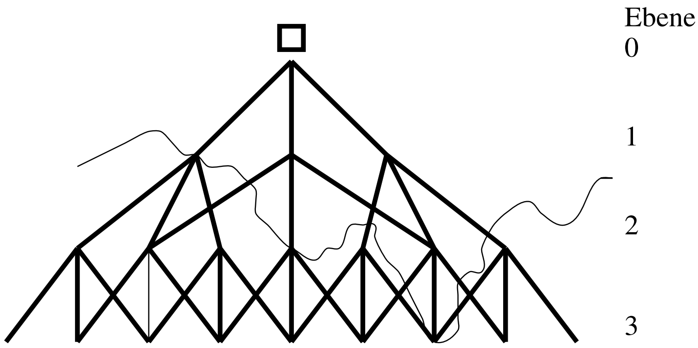

Abbildung 13: Verfeinerungen

MIS: wähle irgendeine Theorie als Ausgangspunkt

repeat lies den nächsten Fakt ein

solange T zu stark ist, wende *contradiction backtracing* an;

solange T zu schwach ist, wende den Verfeinerungsoperator an;

until T ist weder zu schwach noch zu stark bezüglich aller bisher gesehener Fakten;

drucke T aus.

forever

Ein Beispiel für MIS aus dem Bereich natürlicher Zahlen ist:

LO:	0	interpretiert als 0
	X'	interpretiert als Nachfolger von X
	$X \leq Y$	interpretiert als X ist kleiner oder gleich Y
	$plus(X, Y, Z)$	interpretiert als $X + Y = Z$
LH:	Hornformeln	
Fakten:	$0 \leq 0$ ist wahr	
	plus(0,0',0) ist falsch	
	plus(0', 0",0"') ist wahr	
	...	
Theorie:	$X \leq X$	
	$X \leq Y' \leftarrow X \leq Y$	
	plus(X,0,X)	
	plus(X,Y',Z') $\leftarrow$ plus(X,Y,Z)	

Wichtig ist bei diesem Verfahren, daß es ganz offensichtlich – gerade so sind die Bedingungen im Algorithmus formuliert – eine falsche (zu starke oder zu schwache) Theorie verändert und eine einmal gefundene richtige Theorie nicht mehr verändert.

Daß es eine falsche Theorie immer weiter verändert, liegt an der Angemessenheit der Beschreibungssprache für die Hypothesensprache. Daß es einen Grenzwert gibt, zu dem tatsächlich eine richtige Theorie gefunden wird, liegt an der Angemessenheit der Sprachen, der Monotonieeigenschaft der Logik und an der Vollständigkeit des Verfeinerungsoperators. Interessant ist, daß durch die Strukturierung des Hypothesenraums anhand der Verfeinerungsrelation eine Spezialisierung realisiert wird, die der Spezialisierung der G-Menge im Versionenraum entspricht. Wir haben hier aber nicht die starke Einschränkung der Hypothesensprache, wie sie für einen Versionenraum gilt.

9.3 Wahrscheinlich annähernd korrektes Lernen

Wahrscheinlich annähernd korrektes Lernen (*probably approximately correct learning* – **PAC-learning**) stellt wie das Lernen im Grenzwert ein theoretisches Szenario dar, in dem Eigenschaften von Lernverfahren untersucht werden können. Wie schon im vorigen Abschnitt, so ist auch hier die Motivation, so wenig wie möglich von einem Lernverfahren zu fordern und doch noch etwas darüber aussagen zu können. Die gemeinsame Überlegung hinter diesen beiden Paradigmen ist: es ist völlig aussichtslos, ein korrektes und vollständiges Lernverfahren zu fordern, das nach einer bestimmten Menge von Eingaben sicher und prompt das richtige Ergebnis abliefert und dann anhält. Der Unterschied besteht in den vom jeweiligen Paradigma gewählten Abstrichen. Bei der Identifikation im Grenzwert verzichtet man darauf, daß das Verfahren bei der richtigen Lösung anhält. Beim *PAC-learning* schwächt man die Anforderung an die Korrektheit des Lernergebnisses ein. Das Lernergebnis ist nur noch mit einer bestimmten Wahrscheinlichkeit von 1- δ mit einem Fehler von höchstens ϵ richtig. Es wird also nur approximiert, nicht mehr identifiziert. Der Abschwächung bei der Korrektheit stehen aber zwei schwierige Anforderung an das Lernen gegenüber: Das Lernen soll in polynomiell beschränkter Rechenzeit zum Ergebnis kommen, nachdem es eine beschränkte Zahl von Beispielen gesehen hat. Die Beispiele sind in genau der Wahrscheinlichkeitsverteilung, in der tatsächlich Instanzen und Nicht-Instanzen des zu lernenden Begriffs vorkommen. Es wird also eine Stichprobe gegeben. Das Lernergebnis soll die Begriffsdefinition oder Erkennungsfunktion für den Begriff sein.

Hier wird nur das Szenario des *PAC-learning* vorgestellt. Eine kurze, übersichtliche Einführung bietet [Hoffmann, 1991], eine ausführliche Behandlung des Bereiches bietet [Kearns, 1990].

Ein Lernalgorithmus für Begriffe einer Repräsentationsklasse (z.B. Boolsche Funktionen oder Formeln in einer Normalform mit k Termen) erhält Beispiele für einen bestimmten Begriff c aus dieser Repräsentationsklasse. Die Beispiele werden zufällig gewählt, entsprechen aber der "wirklichen", unbekannten Wahrscheinlichkeitsverteilung in der Weise, daß positive Beispiele so wahrscheinlich sind wie Instanzen des Begriffs und negative Beispiele so wahrscheinlich sind wie Nicht-Instanzen des Begriffs. Der Lernalgorithmus erhält außerdem die Parameter δ und ϵ, $\delta < 1$,

$\epsilon < 1$. δ gibt an, mit welcher Wahrscheinlichkeit der Algorithmus den Begriff lernt. ϵ gibt an, wie nahe das Lernergebnis h dem tatsächlichen Begriff c ist, d.h. wieviele Instanzen oder Nicht-Instanzen falsch klassifiziert werden. h klassifiziert Beispiele annähernd korrekt, wenn die Wahrscheinlichkeit e-, daß ein negatives Beispiel als Instanz des Begriffs klassifiziert wird, und die Wahrscheinlichkeit e+, daß ein positives Beispiel als Nicht-Instanz des Begriffs klassifiziert wird kleiner ist als ϵ. Die beiden Parameter schwächen also die Anforderung an die Korrektheit einer gelernten Begriffsdefinition h ab.

Gegeben eine Menge D von Objekten, können wir eine Repräsentationsklasse über D als ein Paar (r, C) definieren, wobei $C \subseteq \{0,1\}^*$ und $r : C \rightarrow 2^D$. $r(c)$ ist ein Begriff und $r(C)$ eine Begriffsklasse. Die gelernte Begriffsdefinition entstammt der Repräsentationsklasse H.

Eine Begriffsklasse C ist **lernbar** durch H, wenn es einen Algorithmus A(δ,ϵ) gibt, der bei einer festen aber beliebigen Wahrscheinlichkeitsverteilung und festen, aber beliebigen ϵ und δ, $\delta < 1$, eine Hypothese $h \in H$ ausgibt, die mit einer Wahrscheinlichkeit größer als 1- δ annähernd korrekt ist, und dann anhält.

Eine Begriffsklasse C ist **polynomiell lernbar** aus Beispielen durch H, wenn Beispiele aus C und H in polynomieller Zeit klassifiziert werden können und der Lernalgorithmus in einer Anzahl von Schritten zum Ergebnis kommt, die sich als Polynom über $1/\epsilon$, $1/\delta$ und $\mid c \mid$ bestimmen läßt. Die Repräsentationsgröße $\mid c \mid$ ist zum Beispiel die Länge einer Repräsentation.

In diesem Szenario kann man nun für bekannte Sprachklassen bzw. ihre Automaten die prinzipielle Lernbarkeit von Begriffen, die in dieser Sprache ausgedrückt sind, untersuchen. Man kann auch die Anzahl der Beispiele errechnen, die man dem Algorithmus geben muß, damit er lernen kann. So sind z.B. Definitionen, die aus gewichteten Attributwerten bestehen, polynomiell lernbar, wenn die Gewichte nicht nur 0 oder 1 beschränkt sind, sondern überall zwischen 0 und 1 liegen. Sind die Gewichte auf 0 oder 1 beschränkt, so sind derartige Definitionen nicht mehr polynomiell lernbar. Das ist deshalb interessant, weil neuronale Netze, die sich gut in das PAC-Paradigma einfügen, gerade durch Gewichtsverschiebungen lernen. Sie bearbeiten also ein polynomiell lernbares Problem. Demgegenüber wählen logik-basierte Verfahren Attribute aus, die sie zur Definition eines Begriffs heranziehen und lassen die anderen Attribute aus der Begriffsdefinition heraus. Sie bearbeiten damit ein schwierigeres Lernproblem.

Michael Kearns verzichtet auf die Forderung, daß derselbe Algorithmus für beliebige Verteilungen von Instanzen funktionieren soll [Kearns, 1990]. Für das schwache Lernen erlaubt er Algorithmen, die auf eine bestimmte Verteilung spezialisiert sind. Kearns beweist, daß ein Algorithmus, der die einheitliche und für positive und negative Beispiele unterschiedliche Wahrscheinlichkeitsverteilung ausnutzt, monotone Boolsche Funktionen lernen kann. Er zeigt einen Algorithmus, der Begriffsdefinitionen in disjunktiver Normalform mit Variablen, die nie mehr als einmal vorkommen,[14]

[14] μ -DNF oder read-once DNF werden solche Formen genannt.

in polynomieller Zeit schwach lernen kann.

Eine weitere Abschwächung im Paradigma des *PAC-learning* schlägt [Pitt, 1990] vor. Danach muß das Lernergebnis nicht mehr die Erkennungsfunktion sein, die in einem bestimmten Repräsentationsformalismus dargestellt wird, sondern kann – in irgendeinem viel mächtigeren Formalismus dargestellt, von einem viel komplexeren Automaten ausgeführt – einfach die wahrscheinlich annähernd korrekte Klassifikation eines neuen Beispiels sein. Allerdings sind auch bei dieser weiteren Abschwächung nicht alle Boolschen Formeln und definite endliche Automaten lernbar.

Eine Abschwächung, die aus dem Paradigma herausführt, brachte bessere Ergebnisse. Es ist die schon oben angeführte Einschränkung des Lernprozesses. Dana Angluin zeigte, daß ein Lernproblem weniger komplex wird, sobald man zuläßt, daß der Algorithmus Fragen an ein Orakel stellt [Angluin, 1988]. Es wird dann nicht nur aus einer Stichprobe von Beispielen gelernt. Schon Äquivalenzfragen (ist die Hypothese äquivalent mit dem Zielbegriff?) machen Formeln in konjunktiver Normalform mit höchstens k Literalen pro Klausel lernbar.

Literatur

[Angluin, 1988] Angluin, D. (1988). Queries and Concept Learning. *Machine Learning*, 2:319–342.

[Angluin und Smith, 1983] Angluin, D. und Smith, C. (1983). Inductive Inference: Theory and Methods. *Computing Surveys*, 15:237–269.

[Barsalou, 1983] Barsalou, L. W. (1983). Ad hoc categories. *Memory and Cognition*, 11(3):211 – 227.

[Boström, 1992] Boström, N. (1992). Eliminating Redundancy in Explanation-Based Learning. In Sleeman, D. und Edwards, P., Hrsg., *Procs. 9th International Workshop on Machine Learning*, Seiten 37 – 42, San Mateo, CA. Morgan Kaufmann.

[Buntine, 1988] Buntine, W. (1988). Generalized Subsumption and Its Applications to Induction and Redundancy. *Artificial Intelligence*, 36:149 – 176.

[Bürckert, 1992] Bürckert, H.-J. (1992). Deduktion, Abduktion, Induktion. *KI*, (3):69 – 70.

[Carey, 1985] Carey, S. (1985). *Conceptual change in childhood*. MIT Press, Boston.

[Clark und Holte, 1992] Clark, P. und Holte, R. (1992). Lazy Partial Evaluation - An Integration of Explanation-Based Generalization and Partial Evaluation. In Sleeman, D. und Edwards, P., Hrsg., *Procs. of 9th International Workshop on Machine Learning*, Seiten 82 –91, San Mateo, CA. Morgan Kaufmann.

[De Jong und Mooney, 1986] De Jong, G. und Mooney, R. (1986). Explanation-Based-Learning: A Alternative View. *Machine Learning*, 2(1):145–176.

[De Raedt, 1991] De Raedt, L. (1991). *Interactive Concept-Learning*. Ph. D. Katholieke Univ. Leuven.

[Dillmann, 1988] Dillmann, R. (1988). *Lernende Roboter - Aspekte maschinellen Lernens*. Springer.

[Dzeroski et al., 1992] Dzeroski, S., Muggleton, S., und Russell, S. (1992). PAC-Learnability of Determinate Logic Programs. *Proceedings of 5th Annual Conference on Computational Learning Theory*, Seiten 128 – 135.

[Emde et al., 1983] Emde, W., Habel, C. U., und Rollinger, C.-R. (1983). The Discovery of the Equator or Concept Driven Learning. In *IJCAI-83*, Seiten 455 – 458, Los Altos, CA. Morgan Kaufman.

[Fargier, 1991] Fargier, H. (1991). Using MOBAL for Security Policy Management - Overview and Remarks. Technical Report AAR-40-1, Alcatel Alsthom Recherche, Marcoussis.

[Fisher, 1987] Fisher, D. H. (1987). Knowledge Acquisition Via Incremental Conceptual Clustering. *Machine Learning*, 2(Douglas H. Fisher):139 – 172.

[Helft, 1989] Helft, N. (1989). Induction as nonmonotonic inference. In *Proceedings of the 1st International Conference on Knowledge Representation and Reasoning*.

[Hoffmann, 1991] Hoffmann, A. (1991). Die Theorie des Lernbaren - ein Überblick. *KI*, (1):7 – 11.

[Jung, 1993] Jung, B. (1993). On Inverting Generality Relations. In Muggleton, S., Hrsg., *Procs. of the 3rd International Workshop on Inductive Logic Programming*, Nummer IJS-DP-6707 in J Stefan Institute Technical Report, Seiten 87 – 101.

[Kearns, 1990] Kearns, M. (1990). *The Computational Complexity of Machine Learning*. ACM Distinguished Dissertation. The MIT Press.

[Keil und Kelly, 1987] Keil, F. und Kelly, M. (1987). Developmental Changes in Category Structure. In Harnad, S., Hrsg., *Categorical Perception*, Kapitel 6, Seiten 491–510. Cambridge University Press.

[Kietz, 1993a] Kietz, J.-U. (1993a). A Comparative Study of Structural Most Specific Generalizations Used in Machine Learning. In Muggleton, S., Hrsg., *Proceedings of the 3rd International Workshop on Inductive Logic Programming*, Nummer IJS-DP-6707 in J. Stefan Institute Technical Reports, Seiten 149 – 164. Also available as Arbeitspapiere der GMD No. 667, 1992.

[Kietz, 1993b] Kietz, J.-U. (1993b). Some Lower Bounds for the Computational Complexity of Inductive Logic Programming. In Brazdil, P., Hrsg., *Machine Learning - Proceedings of ECML-93*, Lecture Notes in Artificial Intelligence, Seiten 115 –123, Berlin, Heidelberg, New York. Springer. Also available as Arbeitspapiere der GMD No. 718, 1992.

[Kietz und Morik, 1993] Kietz, J.-U. und Morik, K. (1993). A Polynomial Approach to the Constructive Induction of Structural Knowledge. *Machine Learning*, 11.

[Kietz und Wrobel, 1991] Kietz, J.-U. und Wrobel, S. (1991). Controlling the Complexity of Learning in Logic through Syntactic and Task-Oriented Models. In Muggleton, S., Hrsg., *Inductive Logic Programming*, Kapitel 16, Seiten 335 – 360. Academic Press, London. Also available as Arbeitspapiere der GMD No. 503, 1991.

[Kodratoff und Ganascia, 1986] Kodratoff, Y. und Ganascia, J.-G. (1986). Improving the generalization step in learning. In Michalski, R. S., Carbonell, J. G., und Mitchell, T. M., Hrsg., *Machine Learning - An Artificial Intelligence Approach*, Kapitel 9, Seiten 215–244. Morgan Kaufman.

[Land, 1983] Land, E. (1983). Verweis ohne Titel in F.J. Varela 1988, Cogn. Science - A Cartography of Current Ideas. *Proc. Natl. Acad. Sci.*, 80.

[Lebowitz, 1986] Lebowitz, M. (1986). Integrated Learning: Controlling Explanation. *Cognitive Science*, 10(2).

[Lebowitz, 1987] Lebowitz, M. (1987). Experiments with Incremental Concept Formation: UNIMEM. *Machine Learning*, 2:103 – 138.

[Lenneberg, 1967] Lenneberg, E. (1967). *Biological Foundations of Language*. New York.

[Michalski, 1986] Michalski, R. (1986). Understanding the Nature of Learning. In Michalski, Carbonell, und Mitchell, Hrsg., *Machine Learning - An Artificial Intelligence Approach*. Morgan Kaufmann, Los Altos, California.

[Michalski und Stepp, 1986] Michalski, R. und Stepp, R. (1986). Conceptual Clustering: Inventing Goal-Oriented Classifications of Structured Objects. In Michalski, R., Carbonell, J., und Mitchell, T., Hrsg., *Machine Learning - An Artificial Intelligence Approach Vol II*, Seiten 471–498. Tioga Publishing Company, Los Altos.

[Michalski, 1983] Michalski, R. S. (1983). A Theory and Methodology of Inductive Learning. In *Machine Learning — An Artificial Intelligence Approach*, Seiten 83 – 134. Morgan Kaufman, Los Altos, CA.

[Michalski und Stepp, 1983] Michalski, R. S. und Stepp, R. E. (1983). Learning from Observation: Conceptual Clustering. In Michalski, R., Carbonell, J., und Mitchell, T., Hrsg., *Machine Learning*, Seiten 331 – 363. Tioga, Palo Alto, CA.

[Michie, 1989] Michie, D. (1989). New Commercial Oppertunities Using Information Technology. In Brauer, F., Hrsg., *Wissensbasierte Systeme*, Seiten 64–71, Berlin, Heidelberg, New York, Tokio. Springer.

[Mitchell, 1982] Mitchell, T. M. (1982). Generalization as Search. *Artificial Intelligence*, 18(2):203 – 226.

[Mitchell, 1985] Mitchell, T. M. (1985). LEAP: A Learning Apprentice for VLSI Design. In *IJCAI 1985, Los Angeles*, Los Altos, California. Morgan Kaufmann Publishers, Inc.

[Morales, 1990] Morales, E. (1990). The Machine Learning Toolkit Database. Deliverable TI-MLT-5.5, The Turing Institute, Glasgow, UK.

[Morik, 1987] Morik, K. (1987). Acquiring Domain Models. *Intern. Journal of Man Machine Studies*, 26:93–104. also appeared in Knowledge Acquisition Tools for Expert Systems, volume 2, J. Boose, B. Gaines, eds., Academic Press, 1988.

[Morik, 1993] Morik, K. (1993). Balanced Cooperative Modeling. *Machine Learning*, 11:217 – 235.

[Morik und Kietz, 1989] Morik, K. und Kietz, J.-U. (1989). A Bootstrapping Approach to Conceptual Clustering. In *Proc. Sixth Intern. Workshop on Machine Learning*.

[Morik et al., 1993] Morik, K., Wrobel, S., Kietz, J.-U., und Emde, W. (1993). *Knowledge Acquisition and Machine Learning - Theory, Methods, and Applications*. Academic Press, London. to appear.

[Mozetic, 1990] Mozetic, I. (1990). Abstractions in Model-Based Diagnosis. In *Procs. of AAI-Workshop on Automatic Generation of Abstractions and Approximations*.

[Muggleton und Buntine, 1988] Muggleton, S. und Buntine, W. (1988). Machine Invention of First-order Predicates by Inverting Resolution. In *Proc. Fifth Intern. Conf. on Machine Learning*, Los Altos, CA. Morgan Kaufman.

[Muggleton und Feng, 1992] Muggleton, S. und Feng, C. (1992). Efficient induction of logic programs. In Muggleton, S., Hrsg., *Inductive Logic Programming*, Kapitel 13, Seiten 281–298. Academic Press, London.

[Murphy und Medin, 1985] Murphy, G. L. und Medin, D. L. (July 1985). The Role of Theories in Conceptual Coherence. *Psychological Review*, 92(3):289 – 316.

[Page und Frisch, 1992] Page, D. und Frisch, A. (1992). Generalization and Learnability: A Case Study of Constrained Atoms. In Muggleton, S., Hrsg., *Inductive Logic Programming*, Jgg. 1, Kapitel 2, Seiten 29 – 62. Academic Press, London, San Diego.

[Piaget, 1977] Piaget, J. (1977). *The development of thought.* Viking Penguin, New York.

[Pitt, 1990] Pitt, L. (1990). Prediction-Preserving Reducability. *Journal of Computing Sciences*.

[Plotkin, 1970] Plotkin, G. D. (1970). A note on inductive generalization. In Meltzer, B. und Michie, D., Hrsg., *Machine Intelligence*, Kapitel 8, Seiten 153–163. American Elsevier.

[Plotkin, 1971] Plotkin, G. D. (1971). A further note on inductive generalization. In Meltzer, B. und Michie, D., Hrsg., *Machine Intelligence*, Kapitel 8, Seiten 101–124. American Elsevier.

[Quine, 1977] Quine, W. V. (1977). Natural Kinds. In Schwartz, Hrsg., *Naming, Necessity, and Natural Kinds*. Cornell Univ. Press.

[Quinlan, 1983] Quinlan, J. R. (1983). Learning Efficient Classification Procedures and Their Application to Chess End Games. In Michalski, R., Carbonell, J., und Mitchell, T., Hrsg., *Machine Learning - An Artificial Intelligence Approach*, Seiten 463 – 482. Tioga, Palo Alto, CA.

[Reimer und Pohl, 1991] Reimer, U. und Pohl, K. (1991). Automatische Wissensakquisition aus Texten. *KI*, Seiten 45 – 51.

[Rieger, 1990] Rieger, A. (1990). Matching Methods for KNowledge-Based Image Interpretation. Arbeitspapier der GMD 485, Gesellschaft fü Mathematik und Datenverarbeitung, GMD Birlinghoven.

[Rosch, 1978] Rosch, E. (1978). Principles of Categorization. In Rosch, E. und Lloyd, B. B., Hrsg., *Cognition and Categorization*, Seiten 27 – 48. Erlbaum, Hillsdale, NJ.

[Rouveirol, 1991] Rouveirol, C. (1991). Semantic Model for Induction of First Order Theories. In *Proceedings 12th IJCAI*, Seiten 685–691. IJCAI, Morgan Kaufmann.

[Rouveirol und Puget, 1990] Rouveirol, C. und Puget, J. F. (1990). Beyond Inversion of Resolution. In Porter, B. und Mooney, R., Hrsg., *Proc. Seventh Intern. Conf. on Machine Learning*, Seiten 122 – 130, Palo Alto, CA. Morgan Kaufmann.

[Scholnick, 1983] Scholnick, E. K., Hrsg. (1983). *New Trends in Conceptual Representation: Challenges to Piaget's Theory?* Lawrence Erlbaum Associates, Hillsdale, NJ.

[Scott, 1983] Scott, P. (1983). Learning: The Construction of a Posteriori Knowledge Structures. In *AAAI-83*, Washington.

[Shapiro, 1981] Shapiro, E. (1981). An Algorithm that Infers Theories from Facts. In *Proc of the seventh IJCAI-81*, Seiten 446–451.

[Shapiro, 1983] Shapiro, E. Y. (1983). *Algorithmic Program Debugging.* ACM Distinguished Doctoral Dissertations. The MIT Press, Cambridge, Mass.

[Simon, 1978] Simon, H. (1978). Acht Vorlesungen über Psychologie. gehalten an der Univ. Hamburg, Fachbereich Psychologie.

[Simon, 1983] Simon, H. (1983). Why Should Machines Learn? In Michalski, R., Carbonell, J., und Mitchell, T., Hrsg., *Machine Learning: An Artificial Intelligence Approach*, Seiten 25–38. Tioga, Palo Alto, CA.

[Spandl und Pitschke, 1991] Spandl, H. und Pitschke, K. (1991). Lernen von Makro-Trajektorien für einen autonomen Roboter. *KI*, Seiten 12 – 16.

[Turing, 1987] Turing, A. (1987). Computing Machinery and Intelligence, in Mind 59, 1950. In Dotzler und Kittler, Hrsg., *Alan Turing - Intelligence Service*. Brinkmann und Bose.

[van Harmelen und Bundy, 1988] van Harmelen, F. und Bundy, A. (1988). Explanation-Based Generalisation = Partial Evaluation. *Artificial Intelligence*, 36:401 – 413.

[Winston, 1987] Winston, P. (1987). *Künstliche Intelligenz*. Addison Wesley, Bonn. German translation of Artificial Intelligence.

[Wirth, 1989] Wirth, R. (1989). Completing Logic Programs by Inverse Resolution. In Morik, K., Hrsg., *Proc. Fourth European Working Session on Learning (EWSL)*, Seiten 239 – 250, London/San Mateo, CA. Pitman/Morgan Kaufmann.

[Wrobel, 1991] Wrobel, S. (1991). Towards a Model of Grounded Concept Formation. In *Proc. 12th International Joint Conference on Artificial Intelligence*, Seiten 712 – 719, Los Altos, CA. Morgan Kaufman.

[Wrobel, 1989] Wrobel, S. (Jan. 1989). Demand-Driven Concept Formation. In Morik, K., Hrsg., *Knowledge Representation and Organization in Machine Learning*, Seiten 289–319. Springer, Berlin, Tokio, New York.

[Zercher, 1991] Zercher, K. (1991). Wissensintensives Lernen von Regeln zur Fehlerdiagnose von Robotermontage. *KI*, Seiten 40 – 44.

Index