

Zur Verwendung nichtmonotoner Inferenztechniken bei der Konfiguration

Gerhard Brewka
GMD
Postfach 13 16
53731 Sankt Augustin
Germany

Torsten Schaub
IRISA
Campus de Beaulieu
35042 Rennes cedex
France

1 Einführung

Ein innerhalb des Bereiches der Künstlichen Intelligenz vorherrschender Ansatz zur Modellierung intelligenten Problemlöseverhaltens geht davon aus, daß für die Problemlösung relevantes Wissen in deklarativer Form repräsentiert und die eigentliche Problemlösung durch geeignete Inferenzverfahren nachgebildet wird. Der weitaus überwiegende Teil unseres Wissens liegt jedoch in einer Form vor, die eine deklarative Repräsentation in klassischen, auf der Prädikatenlogik 1. Stufe beruhenden Repräsentationsformalismen schwierig macht: unser Wissen besteht zu einem wesentlichen Teil aus Regeln, die zwar generelle und typische Zusammenhänge ausdrücken, aber nicht universell gelten, sondern Ausnahmen zulassen. Solche Regeln werden auch Default-Regeln genannt. Das immer wieder zitierte Standardbeispiel ist die Regel "Vögel fliegen (normalerweise)". Diese Regel erlaubt es, von einem bestimmten Vogel, sagen wir Tweety, abzuleiten, daß er fliegen kann. Zusätzliche Information jedoch, etwa, daß es sich bei Tweety um einen Pinguin handelt, kann uns zwingen, diesen plausiblen Schluß zurückzunehmen.

Dieses Beispiel zeigt schon, daß sich das auf Default-Regeln beruhende Schließen in einer wichtigen Hinsicht vom klassischen Schließen unterscheidet: in der klassischen Logik kann eine Erweiterung der Menge der Prämissen nur zu einer Erweiterung der Menge der ableitbaren Formeln führen. Beim Schließen mit Defaults dagegen gibt es Situationen, in denen weitere Information zur Rücknahme von Konklusionen führt, d.h. die Menge der Konklusionen wächst nicht monoton mit der Menge der Prämissen. Diese Eigenschaft hat dem Gebiet des Nichtmonotonen Schließens, in dem es um die Erforschung solcher Inferenztechniken geht, seinen Namen gegeben.

Das Nichtmonotone Schließen hat sich seit Ende der Siebziger Jahre als ein wichtiges eigenständiges Teilgebiet der Künstlichen Intelligenz etabliert. Im Vordergrund der Forschungsarbeiten standen zunächst überwiegend Arbeiten zu den theoretischen Grundlagen des Default-Schließens. Dabei wurden verschiedene formale Ansätze vorgeschlagen, von denen die folgenden sicherlich den meisten Einfluß ausgeübt haben:

- McCarthys Circumscription [22, 23],
- modallogische Ansätze, insbesondere die Autoepistemische Logik [24],
- Reiters Default Logik [26],
- Default-Schließen als inkonistenztolerantes Schließen [25, 5],
- konditionallogische Ansätze [10, 3]

Später hat man sich verstärkt auch mit nichtmonotonen Beweisverfahren auseinandergesetzt. Die erfolgreiche Anwendung des Default-Schließens für realistische Problemstellungen steht jedoch weitgehend noch aus (eine löbliche Ausnahme stellt hier der Bereich der modellbasierten Diagnose dar, in dem die Bedeutung nichtmonotoner Techniken frühzeitig erkannt wurde). Das TASSO-Projekt hat sich deshalb zum Ziel gesetzt, die theoretischen Ansätze zum Default-Schließen für Anwendungen nutzbar zu machen. Das ist zum einen dadurch geschehen, daß mit EXCEPT ein Werkzeug für nichtmonotones Schließen bereitgestellt wurde. Dieses System ist in einem eigenständigen Artikel dieses Buches beschrieben. Zum anderen wurde im Projekt aufgezeigt, wie sich eine bestimmte Klasse von Aufgaben, nämlich Konfigurationsaufgaben, mit Hilfe von Default-Verfahren sinnvoll lösen lassen. Allerdings hat sich im Laufe des Projektes gezeigt, daß sich das Ziel, nichtmonotone Techniken in diesem Sinne für Anwendungen “verfügbarer” zu machen, nur mit Rückwirkung auf die theoretischen Grundlagen erreichen ließ:

- die verfügbaren Techniken, insbesondere Reiters Default Logik, stellen nicht alle für Konfigurationsaufgaben wünschenswerten Sprachmittel zur Verfügung, insbesondere erlauben sie keine adäquate Behandlung von Prioritäten,
- in vielen Fällen führen die Logiken zu unintuitiven Ergebnissen.

Es war deshalb notwendig, auch im Bereich der theoretischen Grundlagen des Default-Schließens intensive Forschungsarbeit zu leisten.

Wir werden im nächsten Abschnitt zeigen, wie sich Techniken des Default-Schließens in der modellbasierten Konfiguration einsetzen lassen. Der darauffolgende Abschnitt gibt einen kurzen Überblick über unsere Arbeiten im Bereich der Theorie der nichtmonotonen Inferenz.

2 Modellbasierte Konfiguration

Konfigurationsaufgaben sind dadurch gekennzeichnet, daß für eine vorgegebene Menge von Zielen und eine vorgegebene Menge von primitiven Komponenten eine bestimmte Auswahl an Komponenten zu finden ist, die eben diese Ziele erfüllen. Ein typisches Beispiel ist etwa das Konfigurieren eines Rechnersystems: es gibt bestimmte vorgegebene Komponenten, die sich zu einem solchen System konfigurieren lassen, und die gewünschte Konfiguration hat bestimmte Ziele zu erfüllen, etwa Bereitstellen einer bestimmten

Speicherleistung, Nichtüberschreiten bestimmter Kosten, etc. Von modellbasierter Konfiguration spricht man dann, wenn ein deklaratives Modell der verfügbaren Komponenten sowie eine deklarative Beschreibung der Ziele vorliegt.

Es lassen sich zwei grundsätzlich unterschiedliche Arten der modellbasierten Konfiguration unterscheiden, die man als *abduktive* bzw. *verfeinerungsbasierte* Konfiguration bezeichnen kann.

Bei der abduktiven Konfiguration enthält die Wissensbasis W Information, die es ermöglicht, aus dem Vorhandensein bestimmter Komponenten das Erfülltsein bestimmter Ziele zu erschließen. Gesucht wird eine minimale Menge von Komponenten K , die folgende Bedingungen erfüllt:

1. $K \cup W$ ist konsistent,
2. $K \cup W \vdash g$ für jedes Ziel g aus der vorgegebenen Zielmenge.

Problematisch an diesem Ansatz ist, daß sehr häufig nicht alle Ziele erfüllt werden können, d.h. daß in vielen Fällen keine Menge K existiert, die die obigen Bedingungen erfüllt. In diesem Fall muß sozusagen per Hand die Menge der Ziele nach und nach verkleinert werden, bis eine Lösung gefunden wird, die maximal viele der Ziele ableitbar macht. Dies ist ein erheblicher Nachteil, und wir haben uns deshalb im wesentlichen auf den zweiten Typ der Konfiguration konzentriert.

Bei der verfeinerungsbasierten Konfiguration ist in der Wissensbasis Wissen darüber repräsentiert, wie bestimmte vorgegebene Ziele erfüllt werden können. Dabei werden Grobziele verfeinert zu Feinzielen. Diese Verfeinerung geht weiter, bis man bei primitiven Komponenten anlangt. Im Beispiel des Diagrammgenerators etwa könnte ein Grobziel lauten:

Vorliegende Daten graphisch darstellen

Dieses Grobziel ließe sich etwa verfeinern durch die folgenden (alternativen) feineren Ziele:

Daten durch Kuchendiagramm darstellen oder
Daten durch Säulendiagramm darstellen oder

...

Die Verfeinerung der Ziele wird bis hin zur exakten und ausführbaren Beschreibung der Darstellungsform vorgenommen. Zusätzlich können Regeln spezifiziert werden, die in bestimmten Situationen gewisse Darstellungsformen ausschließen oder empfehlen.

Default-Regeln sind für die Konfiguration aus zwei Gründen relevant:

1. Ziele können zum einen unterspezifiziert sein, das heißt sie können bestimmte Auswahlmöglichkeiten offen lassen. Häufig haben aber nicht alle diese möglichen Lösungen die gleiche Qualität. In diesem Fall können wir Defaults benutzen, um den besseren Lösungen Priorität vor den zwar auch möglichen, aber nicht so guten Lösungen zu geben. Z.B. kann die Wissensbasis die Information enthalten

Benutze für Zeitreihendarstellung Kurven- oder Säulendiagramm

Durch die Default-Regel

Wenn Zeitreihendarstellung dann normalerweise Kurvendiagramm

wird die bessere erste Alternative bevorzugt und ausgewählt, falls das konsistent möglich ist. Nur wenn aufgrund von Defaults höherer Priorität abgeleitet wurde, daß die Lösung kein Kurvendiagramm beinhalten soll, wird ein Säulendiagramm gewählt.

2. Ziele können widersprüchlich sein. In diesem Fall kann man dadurch, daß man die Ziele selbst als Defaults formuliert, zu Lösungen kommen, die zwar nicht alle, aber wenigstens die wichtigsten Ziele erfüllen. Enthält etwa die Wissensbasis die Information:

Benutze für Strukturvergleich Kuchendiagramm.

Benutze kein Kuchendiagramm bei mehr als 5 Datenpunkten.

so erhält man bei einem Strukturvergleich mit mehr als 5 Datenpunkten einen Widerspruch. Der kann dadurch behoben werden, daß man eine oder beide Regeln zu Defaults macht. Die Priorität des jeweiligen Defaults entscheidet darüber, welche Lösung im Konfliktfall vorgeschlagen wird.

In [20] wird der verfeinerungsbasierte Ansatz ausführlicher beschrieben. Dort wird auch diskutiert, wie das Nichterfülltsein eines bestimmten Teilziels festgestellt werden kann. Man betrachte z.B. das Teilziel

Stelle Daten durch ein Balkendiagramm dar.

Wenn für dieses Teilziel nur die beiden Verfeinerungen

Wähle gestuftes Balkendiagramm oder

Wähle gruppiertes Balkendiagramm

zur Verfügung stehen, die erste jedoch aus Platzmangel und die zweite an der Lesbarkeit scheitert, dann läßt sich obiges Teilziel nicht realisieren. Klassische Suchverfahren würden in diesem Falle automatisch das Teilziel aufgeben und im Suchraum zurückgehen. In unserem logischen Ansatz verwenden wir dazu die Aussage:

*Wenn die Tabelle durch ein Balkendiagramm dargestellt werden soll,
dann realisiere sie als gestuftes Balkendiagramm oder als gruppiertes Balkendiagramm.*

Somit werden bei dem verfeinerungsbasierten Ansatz Disjunktionen benötigt, um nicht erfüllbare Teilziele zu erkennen. Dies wirkt zwar technisch aufwendig, hat aber den Vorteil,

daß bekannt ist, welche Ziele miteinander in Konflikt stehen, wenn nicht alle erfüllbar sind. Die üblichen Prioritäten können benutzt werden, um wichtigere Ziele zu bevorzugen.

Insgesamt sollte in diesem Abschnitt klar geworden sein, daß die Möglichkeit, Defaults in der Beschreibungssprache verwenden zu können, einen immensen Vorteil bei der Lösung von Konfigurationsaufgaben darstellt.

3 Beiträge zur Theorie der Nichtmonotonie

Wir haben bereits in der Einleitung erwähnt, daß die verfügbaren Formalisierungen nicht-monotoner Inferenz für die TASSO-Zielsetzungen in verschiedener Hinsicht unbefriedigend waren. In diesem Abschnitt wollen wir erläutern, welche Probleme auftraten, und wie sie durch Modifikationen bzw. Erweiterungen der Logiken behoben wurden.

Unsere theoretischen Arbeiten beruhen im wesentlichen auf zwei Ansätzen: der Reiter-schen Default Logik [26] und dem Ansatz der bevorzugten Teiltheorien, wie er in [5, 6] beschrieben wurde. Bei beiden Ansätzen handelt es sich um konsistenzbasierte Formalismen, bei denen bei der Definition der ableitbaren Formeln explizit auf den Begriff der Konsistenz zurückgegriffen wird. Wir wollen zunächst die Default Logik näher erläutern.

Bei der Default Logik werden Default-Regeln als Inferenzregeln mit einem zusätzlichen Konsistenztest repräsentiert. Defaults werden dabei folgendermaßen notiert:

$$A : B/C$$

Diese Regel läßt sich wie folgt interpretieren: “wenn A abgeleitet wurde und B konsistent ist mit der Menge der abgeleiteten Formeln, dann leite B ab”. Default Theorien bestehen aus einer Menge klassischer Formeln W , die das sichere Wissen repräsentiert, und einer Menge D von Defaults obiger Art. Default Theorien generieren sogenannte Extensionen, das sind Mengen von Formeln, die akzeptable Überzeugungsmengen repräsentieren. Eine Extension E muß folgende Eigenschaften erfüllen:

1. das sichere Wissen W ist in E enthalten,
2. E ist abgeschlossen im Sinne der klassischen Logik,
3. alle “anwendbaren” Defaults wurden angewendet, d.h. $A : B/C \in D$, $A \in E$, $\neg B \notin E$ impliziert $C \in E$.
4. alle Formeln in E müssen auf nicht-zirkuläre Weise aus W und D herleitbar sein.

Es stellt sich heraus, daß diese Eigenschaften von Extensionen nur mithilfe einer Fixpunktdefinition formal beschrieben werden können. Dabei kann es vorkommen, daß eine Default Theorie mehrere (oder auch gar keine) Extensionen besitzt.

Der Vorteil der Default Logik liegt zum einen in ihrer Ausdrucksmächtigkeit, zum anderen in der intuitiven Darstellung der Defaults, die klassischen Inferenz- oder Expertensystemen mit ihren monotonen Inferenzregeln recht nahe kommt. Auf die Mängel der Default Logik werden wir in Kürze eingehen.

Der Ansatz der bevorzugten Teiltheorien (preferred subtheories) erweitert Pooles Theorie des Default-Schließens [25] um Prioritäten. Dabei werden Defaults als klassische Formeln dargestellt, die aber zusätzlich mit einem Grad an Relevanz bzw. einer Priorität versehen sind. Treten nun Konflikte zwischen Formeln auf, etwa weil eine Formel sicheres, die andere nur tentatives Wissen beschreibt, dann "gewinnt" die Formel höherer Priorität und setzt die andere Formel außer Kraft. Die Prioritäten können in einer einfacheren Variante durch Zuordnung zu einer Prioritätsstufe ausgedrückt werden, in einer allgemeineren Version können beliebige partielle Ordnungen dafür verwendet werden.

Dieser Ansatz stellt, was die Repräsentation von Defaults angeht, einen interessanten Spezialfall von Reiters Logik dar: einer Formel A entspricht ein Default der Form $true : A/A$. Andererseits gibt es für die einfache Darstellung von Prioritäten keine Entsprechung in der Default Logik.

Ausgehend von diesen beiden Ansätzen wurden eine Reihe von Problemen theoretischer Art untersucht. Dabei lag der Schwerpunkt auf folgenden Bereichen:

- Varianten der Default Logik,
- Semantiken für Default Logiken,
- Behandlung von Prioritäten,
- Prognostizieren mit Defaults,
- "Meta"-Theorie des Default-Schließens,
- Berechnungsverfahren, EXCEPT

Wir werden zu jedem dieser Punkte kurz die wesentlichen Projektergebnisse erläutern. Da auf das System EXCEPT, das die entwickelten Berechnungsverfahren zur Verfügung stellt, in einem gesonderten Artikel eingegangen wird, werden wir uns bei Beschreibung des letzten Punktes kurz fassen.

3.1 Varianten der Default Logik

Es stellt sich heraus, daß eine ganze Reihe von Problemen bei der Repräsentation von Wissen mit der Reiterschen Logik auftreten:

1. die Existenz von Extensionen, also alternativen Mengen von plausiblen Überzeugungen, ist nicht gewährleistet,
2. teils werden intuitiv erwartete Konklusionen nicht vorgenommen, teils unplausible Schlüsse gezogen,
3. die Verwendung abgeleiteter Formeln als Prämissen ändert die erzielten Resultate,
4. Prioritäten lassen sich nur schwer repräsentieren

5. die Spezifität von Defaults wird nicht berücksichtigt.

Diese Probleme haben zur Entwicklung verschiedener verbesserter Varianten der Default Logik geführt. Die kumulative Default Logik, CDL [7], verwendet als atomare Elemente nicht Propositionen, wie die klassische Logik, sondern komplexere Strukturen, sogenannte Assertionen, die aus einer Proposition zusammen mit einer Menge von Gründen bestehen, die zu der Annahme der Proposition geführt haben. Dadurch lassen sich einige der obengenannten Probleme lösen. Allerdings werden auch in CDL einige intuitiv erwartete Ableitungen nicht vorgenommen.

Um auch dieses Problem zu lösen, wurde ein Rahmen zur Definition kumulativer Default Logiken entwickelt, bei dem die Defaults selbst zu Elementen der logischen Sprache werden [8]. Damit lassen sich neue Defaults aus den vorgegebenen ableiten und so die erwarteten Resultate erzielen. Insbesondere können Probleme, die auf unzureichender Behandlung von Fallunterscheidungen beruhen, in diesem Ansatz behoben werden.

Die zweite im Rahmen von TASSO entwickelte Variante ist die bedingte Default Logik, BDL [30, 31, 11]. Die Grundidee liegt bei ihr darin, eine Extension mit einer Menge von Randbedingungen zu versehen, die einer Menge von Gründen für die in der Extension enthaltenen Überzeugungen entspricht. Damit wird der kontextsensitive Charakter der Default Logik explizit gemacht und zwischen der eigentlichen Extension und den ihr zugrundeliegenden Konsistenzannahmen unterschieden. Viele wichtige Eigenschaften gelten in der BDL in größeren Teilklassen als in Reiters Default Logik. Zum Beispiel werden viele Eigenschaften von gutartigen aber stark eingeschränkten Defaulttheorien auf generelle Defaulttheorien verallgemeinert. In [31, 33] wird eine Erweiterung der BDL um Prioritäten und vordefinierte Randbedingungen vorgestellt. Auf eine weitere Variante der Default Logik, die explizite Prioritäten behandeln kann, werden wir in Abschnitt 3.3 eingehen.

Die Beziehungen zwischen den verschiedenen Varianten der Default Logik, nämlich zwischen "klassischer" Default Logik [26], einer Variante der Default Logik von Łukasiewicz [21], der CDL und der BDL, wurden in [32, 31, 11] eingehend untersucht. Insbesondere wurden dabei Kriterien angegeben, die aussagen unter welchen Bedingungen die verschiedenen Varianten übereinstimmen. Grob gesprochen, hat es sich dabei gezeigt, daß die Reiterische Default Logik und die Variante der Default Logik von Łukasiewicz [21] auf der einen Seite, sowie die CDL [7] und die BDL auf der anderen Seite, jeweils enger miteinander verwandt sind.

Die Untersuchung der in der Default Logik auftretenden Phänomene und die damit verbundene Weiterentwicklung der Default Logik und ihrer Varianten gipfelt in der Entwicklung der kontextuellen Default Logik, KDL [2]. Dieser Ansatz stellt ein einheitliches syntaktisches Rahmenwerk für Default Logiken zur Verfügung, indem er verallgemeinerte Default-Regeln verwendet und Extensionen mit Kontexten versieht. Neben der Einbettung bekannter Defaultlogiken, wie der von Reiter, von Łukasiewicz [21], der CDL als auch der BDL, erlaubt der Ansatz obendrein auch die Formulierung traditioneller Ansätze, wie den der Annahme der Weltabgeschlossenheit und den der Beweisbarkeit in Prädikatenlogik erster Stufe. Zudem wird es durch eine äußerst homogene Codierung möglich, diese Ansätze in beliebiger Form zu kombinieren. Ein weiterer Vorteil dieses Ansatzes liegt in

der Möglichkeit verschiedene Defaultlogiken innerhalb eines einheitlichen syntaktischen Rahmens zu vergleichen. Insbesondere hat der Ansatz aufgezeigt, daß sich existierende Defaultlogiken vornehmlich durch ihre Behandlung explizit oder implizit vorhandener Kontexte unterscheiden.

3.2 Semantiken für Default Logiken

Die im letzten Abschnitt beschriebene Weiterentwicklung der Reiterschen Default Logik wurde von einer intensiven semantische Analyse der Default Logiken und der in ihnen auftretenden Phänomene begleitet.

Zunächst wurde eine semantische Charakterisierung der BDL und CDL, die Fokussierte-Modelle Semantik entwickelt [29, 31]. Im Zusammenhang mit der BDL stellt sich die Grundidee der Fokussierte-Modelle Semantik wie folgt dar: Eine Extension und ihre Randbedingungen wird mit Hilfe von strukturierten Klassen von Modellen beschrieben. Dazu wird eine Teilklasse — die Klasse der sogenannten fokussierten Modelle — ausgezeichnet, die zusätzlich zur Menge der in einer Extension enthaltenen Schlußfolgerungen auch die diesen Schlußfolgerungen zugrunde liegenden Annahmen erfüllt. Dadurch entspricht dann die Klasse der fokussierten Modelle den Randbedingungen einer Extension. Dahingegen wird die Extension selbst durch sämtliche Modelle einer bestimmten Modellklasse charakterisiert.

Die Fokussierte-Modelle Semantik hat sich als ein Instrument zur semantischen Analyse der BDL und CDL bewährt. Obwohl sie in einem gewissen Sinne auch auf Reiters Default Logik übertragbar ist, stellt sie dennoch keine einheitliche Semantik für Default Logiken dar. Beispielsweise ist sie nicht auf die von Lukaszewicz in [21] entwickelte Default Logik übertragbar.

Es ist nun im Rahmen von TASSO erstmals gelungen, einen einheitlichen semantischen Rahmen für Default Logiken mit Hilfe von Kripke-Strukturen zu entwickeln [1, 31]. Dieser semantische Ansatz erlaubt durch seinen universellen Charakter den Vergleich und das Studium von Default Logiken innerhalb eines einheitlichen Rahmens. Außer seiner einzigartigen Allgemeinheit, vermeidet der Ansatz insbesondere etliche Schwierigkeiten herkömmlicher Semantiken, die zudem auf spezielle Default Logiken zugeschnitten sind. Insbesondere stellt diese Semantik die erste pure modelltheoretische Semantik für Lukaszewicz' Variante der Default Logik dar [21].

Kripke-Strukturen [4] werden traditionell zur Charakterisierung modaler und temporaler Logiken verwendet, da sie es erlauben, Beziehungen zwischen Modellen oder sogenannten Welten herzustellen. Kripke-Strukturen haben eine ausgezeichnete Welt und eine Menge von Welten, die von ihr aus "erreichbar" sind. Die Intuition hinter dieser Semantik ist dann sehr natürlich und leicht zu verstehen: Die ausgezeichnete Welt charakterisiert den aktuellen Stand unserer Überzeugungen und damit die Extension, wohingegen die sie umgebenden Welten zusätzlich zu unseren Überzeugungen auch unsere Annahmen verkörpern, die zu diesen Überzeugungen geführt haben. Formal wird dies durch eine von Default-Regeln induzierte strikte partielle Ordnung auf Klassen von Kripke-Modellen realisiert, die mit Hilfe der Modallogik K beschrieben wird.

3.3 Behandlung von Prioritäten

Bei dem oben skizzierten Ansatz zur Verwendung von Defaults für die Konfiguration sind verschiedene Lösungen der Konfigurationsaufgabe in verschiedenen Theorien repräsentiert, die durch die Anwendung einer maximalen Menge von "konsistent" anwendbaren Defaults erzeugt wurden. Im allgemeinen Fall kann es sehr viele solche Lösungsvorschläge geben, wenn Konflikte zwischen den vorhandenen Defaults existieren. Häufig sind nicht alle diese Lösungen gleich gut, und man kann das Erzeugen von weniger plausiblen Lösungen dadurch vermeiden, daß man Prioritäten zwischen Defaults einführt und im Konfliktfall nur die Defaults höherer Priorität anwendet. Eine adäquate Behandlung von Prioritäten ist deshalb eine wichtige Voraussetzung für den Einsatz von Defaults in der Konfiguration.

Eine solche Behandlung von Prioritäten bietet der Ansatz der bevorzugten Teiltheorien (preferred subtheories). Hier wird Default Schließen dadurch realisiert, daß aus der insgesamt verfügbaren Information auf geeignete Weise maximal konsistente Teilmengen ausgewählt werden, die dann zur Bestimmung von akzeptablen Mengen von Überzeugungen verwendet werden können. Die Auswahl wird, wie oben erwähnt, dadurch beschränkt, daß die verfügbare Information vom Benutzer mit Prioritäten versehen wird.

In einem stufenbasierten Ansatz werden alle Defaults in eine bestimmte Prioritätsstufe einsortiert. In einem allgemeineren, auf partiellen Ordnungen beruhenden Ansatz können beliebige Präferenzrelationen zwischen den Defaults spezifiziert werden.

Ein wesentlicher Teil unserer Arbeit konzentrierte sich auf den Entwurf und die Bereitstellung von Algorithmen, die solche Prioritäten geeignet verarbeiten können. Zunächst wurden die von Junker und Konolige entwickelten Verfahren zur Berechnung von Default Logik Extensionen [19] weiterentwickelt und für den stufenbasierten Ansatz einsetzbar gemacht [12]. Interessanterweise führt die Behandlung von Prioritätsstufen nicht zu einer Verminderung der Effizienz des Verfahrens, sondern die zugrundeliegenden Datenstrukturen (Netze in Begründungsverwaltungssystemen (engl. Truth Maintenance Systemen)) werden dabei sogar kleiner.

Für viele Anwendungsgebiete sind Prioritätsstufen allerdings zu starr und inflexibel, etwa bei Vererbungshierarchien, Diagnose und insbesondere auch der Konfiguration. Sie zwingen den Benutzer oft dazu, unerwünschte Prioritäten einzuführen. Deshalb sind partielle Ordnungen zwischen Defaults von erheblicher Bedeutung. Wie sich Truth-Maintenance-basierte Methoden und Enumerationsverfahren einsetzen lassen, um partielle Präferenzordnungen auf Defaults zu behandeln, wurde in [18] gezeigt. Diese Verfahren sind in EXCEPT verfügbar gemacht worden.

Es stellte sich heraus, daß der preferred subtheory Ansatz nicht nur für Konfiguration, sondern auch für modellbasierte Diagnose gut geeignet ist [12, 14]. Bei der modellbasierten Diagnose wird das korrekte Verhalten der Bauteile eines Systems formal beschrieben. Zusätzlich werden Korrektheitsannahmen bezüglich der Bauteile gemacht. Die als fehlerhaft diagnostizierten Komponenten sind gerade die, für die eine solche Annahme zum Widerspruch führt. Defaults mit Prioritäten bieten die Möglichkeit, Korrektheitsannahmen unterschiedlich zu gewichten und damit die unterschiedliche Zuverlässigkeit verschiedener Komponenten zu repräsentieren.

Die Prioritätsbehandlung der bevorzugten Teiltheorien läßt sich auch auf die Default Logik übertragen, die ja, was die Darstellung von Defaults angeht, mächtiger ist: wie bereits erwähnt entsprechen Defaults bei den bevorzugten Teiltheorien Reiter-Defaults der Form $true : p/p$. Eine Variante der Default Logik, die die Behandlung von expliziten Prioritäten zuläßt, ist in [9] beschrieben. Dabei werden die Defaults partiell geordnet und nur solche Extensionen erzeugt, die diese Ordnung berücksichtigen. Damit verbindet diese Variante die Ausdrucksmächtigkeit der Default Logik mit der einfachen Prioritätsbehandlung der bevorzugten Teiltheorien. In der zitierten Arbeit wird auch gezeigt, wie die Prioritäten definiert werden können, die man benötigt, um spezifischere Regeln allgemeineren vorziehen zu können.

3.4 Prognostizieren mit Defaults

Dem Pooleschen THEORIST-Ansatz zur Prognose liegt eine skeptische Sichtweise zugrunde: eine Formel wird dann prognostiziert, wenn sie in jeder Extension, d.h. in jeder um eine maximal konsistente Menge von möglichen Annahmen erweiterten Weltbeschreibung, gilt.

Es stellt sich heraus, daß das für THEORIST von Poole in Prolog implementierte System Prognosen nur für einen eingeschränkten Bereich korrekt durchführt.

In [34] wurde eine diese Schwäche behebende Weiterentwicklung des Systems vorgestellt, deren theoretische Grundlagen in [35] beschrieben werden. Als wichtigstes Ergebnis stellte sich dabei die Formulierung eines neuen Begriffs, den des inkompatiblen Szenarios dar, der sich von dem bekannten Begriff des inkonsistenten Szenarios unterscheidet und für eine korrekte Durchführung einer Prognose in dem vollständigen Ansatz unerlässlich ist. Mit Hilfe dieses neuen Begriffs konnte in [35] die Korrektheit des in [34] vorgestellten Algorithmus bewiesen werden.

3.5 Explizite Kontrolle der Einführung und Rücknahme von Annahmen

Die Anwendbarkeit nicht-monotoner Logiken wird derzeit durch die Vielzahl sehr unterschiedlicher und mathematisch komplizierter Formalismen stark erschwert. Im Rahmen der Dissertation von U. Junker [16] wird gezeigt, daß eine einfachere Vorgehensweise ausreichend ist:

- Der Benutzer kennzeichnet einige Regeln als rücknehmbar, was im System durch Aufheben einer neuen Annahme in die Vorbedingung der Regel erzielt wird.
- Eine Lösung ist eine zusammenhängende Menge von Merkmalen, die alle Randbedingungen (engl. Constraints) und Anforderungen erfüllen und z.B. eine Beschreibung einer Graphik, also eines komplexeren Objekts darstellen.

Das technische Kernproblem, das sich nun stellt und durch die verschiedenen nicht-monotonen Logiken z.T. sehr umständlich gelöst wird, ist die Frage, welche Annahmen

bei einem Konflikt zwischen Regeln zurückgenommen werden sollen.

Der in TASSO entwickelte Ansatz der bevorzugten Teiltheorien löst das angesprochene Kontrollproblem mit Hilfe der vorgegebenen Prioritäten zwischen Annahmen. Im Fall eines Konflikts wird einfach eine schwächste Annahme aus dem Konflikt zurückgenommen. Somit werden einige Annahmenmengen aufgrund der Prioritäten bevorzugt, was sich auch mit Hilfe einer Präferenzrelation zwischen Annahmenmengen ausdrücken läßt [18].

Auch die Defaults in der Default Logik und die modalen Formeln in autoepistemischer Logik lassen sich durch herkömmliche Regeln und zusätzliche Annahmen ausdrücken. Beide Logiken verfügen jedoch über spezielle und in den Ausgangsdefinitionen stark versteckte Mechanismen, um einige Annahmenmengen zu bevorzugen. In [16] wurden diese Mechanismen explizit gemacht und auf sehr einfache Weise ausgedrückt. Die Einführung und Rücknahme von Annahmen wird über zwei Spezialprädikate “anwendbar” und “blockiert” geregelt. Das erste Prädikat ist ein positiver Indikator für das Hinzunehmen einer Annahme, das zweite ein positiver Indikator für das Rücknehmen. Um bevorzugte Annahmenmengen zu erhalten, geht man schrittweise vor. Man startet mit einer beliebigen Annahmenmenge und modifiziert diese wie folgt:

- Füge eine Annahme hinzu, die anwendbar und nicht blockiert ist.
- Nimm eine Annahme zurück, die blockiert oder nicht anwendbar ist.

Durch diese Operationen wird wieder eine Präferenzrelation zwischen Annahmenmengen definiert. Bevorzugte Lösungen sind solche, die nicht verbesserbar sind. In [15] wurde nachgewiesen, daß sich mit dieser einfachen Methode die gleichen Ergebnisse wie in Default Logik oder Autoepistemischer Logik erzielen lassen.

Damit ist ein einheitlicher und anwenderfreundlicherer Rahmen entstanden, in dem sich verschiedene Strategien zum Behandeln von Konflikten, Abhängigkeiten und Präferenzen zwischen Annahmen behandeln lassen. Die dabei verwendeten Spezialprädikate werden auch bei der Darstellung der Default-Regeln in EXCEPT benutzt.

3.6 Das Default-Reasoning-System EXCEPT

EXCEPT ist ein Default Reasoning System, das dem Benutzer die Möglichkeit bietet, Defaults mit partiell geordneten Prioritäten zu spezifizieren. Das System ist im Laufe des Projektes mehrfach rekonzipiert und reimplementiert worden. Die derzeit aktuelle Version baut, im Gegensatz zu früheren Versionen, auf einem Doyle'schen TMS anstatt auf einem ATMS auf. Damit kann besser auf den aktuellen Kontext des Problemlöseprozesses fokussiert werden. Das TMS übernimmt dabei eine Vielzahl von Funktionen:

1. Kontextverwaltung,
2. Behandlung negativer Annahmen,
3. Einführung und Rücknahme von Default-Annahmen,

4. Konfliktbestimmung,
5. Verwaltung des Suchraums für weitere Lösungen.

Um die dabei auftretenden technischen Probleme zu lösen, wurde die gesamte Systemarchitektur von EXCEPT mehrfach neu gestaltet, wobei insbesondere die TMS-Aufgaben der Kontextverwaltung und der Annahmeneinführung vollständig getrennt wurden. Es wurden folgende Abstraktionsschichten eingeführt:

1. Einführung und Rücknahme von Annahmen
2. Interface-Funktionen
3. Bestimmung eines minimalen Herbrandmodells (MMMS)
4. Instantiierung von nicht erfüllten Regeln

Die unterste Schicht entspricht der üblichen Datenverwaltung und dem Matching von Regeln. Die zweite Schicht realisiert das System MMMS, das den Kontext, d.h. ein minimales Herbrandmodell verwaltet. Hierzu müssen sowohl anwendbare Regeln ausgeführt, als auch nicht mehr anwendbare Regeln zurückgenommen werden. Weiterhin bestimmt das MMMS Konflikte zwischen Annahmen, falls der aktuelle Kontext inkonsistent ist. Verschiedene Interface-Funktionen benutzen das MMMS für Konsistenztests und zum Deduzieren von Kontrollbedingung für Annahmen. Diese sind wiederum die Primitiva für die Algorithmen zur Kontrolle der Einführung und Rücknahme von Annahmen. Diese Algorithmen lassen sich direkt aus einigen der im Abschnitt über die explizite Kontrolle von Annahmen vorgestellten Präferenzrelationen gewinnen.

Diese neue Architektur gewährleistet damit eine einfache Implementierung der Methoden zur Einführung und Rücknahme von Annahmen. Da dies nicht tief versteckt im Code gemacht wird, lassen sich diese Algorithmen leicht austauschen, so daß es auf einfache Weise möglich ist, die dem System zugrundeliegende nichtmonotone Logik auszutauschen.

Für eine ausführlichere Darstellung von EXCEPT verweisen wir auf den in diesem Band enthaltene Artikel von Ulrich Junker. Die im Projekt mit dem System entwickelte Anwendung im Bereich der Graphik-Konfigurierung ist in dem Artikel von Rüdiger Kolb beschrieben.

Literatur

- [1] P. Besnard und T. Schaub. Possible Worlds Semantics for Default Logics. *Fundamenta Informaticae*, 20(1–2), 1993.
- [2] P. Besnard und T. Schaub. A context-based framework for default logics. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Seite 406–411, 1993.

- [3] C. Boutilier, Conditional Logics for Default Reasoning and Belief Revision, University of British Columbia, TR 92-1, 1992
- [4] K. Bowen. Model Theory for Modal Logics. Reidel, Synthese Library, Dordrecht, 1979.
- [5] G. Brewka, Preferred Subtheories: An Extended Logical Framework for Default Theories, Proc IJCAI-89, Detroit, 1989
- [6] G. Brewka, Bevorzugte Teiltheorien: Wissensrevision in einem Ansatz zum Default-Schließen. Kognitionswissenschaft 1 (1990), Springer Verlag
- [7] G. Brewka, Cumulative Default Logic - In Defense of Nonmonotonic Inference Rules, Artificial Intelligence 50, 1991
- [8] G. Brewka, A Framework for Cumulative Default Logics, ICSI TR-92-012, 1992
- [9] G. Brewka, Adding Priorities and Specificity to Default Logic. Zur Veröffentlichung eingereicht.
- [10] J. P. Delgrande, An Approach to Default Reasoning Based on a First-Order Conditional Logic, Artificial Intelligence 36, 1988
- [11] J. P. Delgrande, W. K. Jackson und T. Schaub. Alternative Approaches to Default Logic. Artificial Intelligence. Erscheint 1994.
- [12] U. Junker, Prioritized Defaults: Implementation by TMS and Application to Diagnosis, TASSO-Report Nr. 21, GMD, Marz 1991, ebenso in: Proc. IJCAI 91, Sydney, Australia
- [13] U. Junker, The EXCEPT II Default Reasoning System - Preliminary Report, TASSO-Report Nr. 23, GMD, Juni 1991
- [14] U. Junker, Generating Diagnoses by Prioritized Defaults, erscheint in: Proc. Second International Workshop on Principles of Diagnosis, Milano, Oct. 1991
- [15] U. Junker: Relationships between Assumptions. Dissertation, Universität Kaiserslautern, Kaiserslautern, 1992.
- [16] U. Junker: Controlling the Selection and Retraction of Assumptions. in: Proc. ECAI Workshop on Theoretical Foundations of Knowledge Representation and Reasoning, Wien, 1992.
- [17] U. Junker: Preferring Diagnoses Using a Partial Order on Assumptions, Special Issue on Model-Based Diagnosis, Annals of Mathematics and Artificial Intelligence, 1993
- [18] U. Junker, G. Brewka, Computing Partially Ordered Defaults, in Proc. European Conference on Quantitative and Qualitative Approaches to Uncertainty, Marseille, Okt. 91, Springer Verlag, 1991

- [19] U. Junker, K. Konolige: Computing the Extensions of Autoepistemic and Default Logics with a TMS, Proceedings AAAI 1990, Boston
- [20] U. Junker und J. Paulokat: Logic-Based Methods for Design. draft, 1992.
- [21] W. Łukaszewicz. Considerations on Default Logic — An Alternative Approach. Computational Intelligence, 4:1–16, 1988.
- [22] McCarthy, John, Circumscription - A Form of Commonsense Reasoning, Artificial Intelligence 13, 1980
- [23] McCarthy, John, Applications of Circumscription to Formalizing Commonsense Knowledge, Artificial Intelligence 28, 1986
- [24] Moore, Robert C., Semantical Considerations on Nonmonotonic Logic, Artificial Intelligence 25, 1985
- [25] Poole, David, A Logical Framework for Default Reasoning, Artificial Intelligence 36 (1988) 27-47.
- [26] Reiter, Raymond, A Logic for Default Reasoning, Artificial Intelligence 13 (1980) 81-132.
- [27] T. Schaub. Nichtmonotone Logiken und ein Default-Beweiser. TASSO-Report 6, FG Intellektik, FB Informatik, TH Darmstadt, July 1990.
- [28] T. Schaub. Nichtmonotone Logiken und ein Default Beweiser. Diplomarbeit, FG Intellektik, FB Informatik, TH Darmstadt, May 1990.
- [29] T. Schaub. Assertional Default Theories: A Semantical View. In J. A. Allen, R. Fikes und E. Sandewall (Hg.), Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning, San Mateo, CA, Seite 496–506. Morgan Kaufmann Publishers Inc., April 1991.
- [30] T. Schaub. On Commitment and Cumulativity in Default Logics. In R. Kruse (Hg.), Proceedings of European Conference on Symbolic and Quantitative Approaches to Uncertainty, Seite 304–309. Springer Verlag, 1991.
- [31] T. Schaub. Considerations on Default Logics. Doktorarbeit, Technische Hochschule Darmstadt, Alexanderstraße 10, D-6100 Darmstadt, November 1992.
- [32] T. Schaub. On Constrained Default Theories. In B. Neumann (Hg.), Proceedings of the European Conference on Artificial Intelligence, Seite 304–308. John Wiley & sons, 1992.
- [33] T. Schaub. Variations of constrained default logic. In M. Clarke, editor, Proceedings of European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, Springer Verlag, 1993. Erscheint demnächst.

- [34] M. Thielscher. Prognostizieren im Poole'schen Ansatz – Arbeiten mit THEORIST. TASSO-Report 18, FG Intellektik, FB Informatik, TH Darmstadt, January 1991.
- [35] M. Thielscher. On prediction in Theorist. *Artificial Intelligence (Research Note)*, 60(2):283–292, 1993.