UNIVERSITÄT DORTMUND
FACHBEREICH INFORMATIK

LEHRSTUHL VIII
KÜNSTLICHE INTELLIGENZ

Planen von Aktionen und Reaktionen

Ein einführender Überblick

LS–8 Report 7

Joachim Hertzberg

Dortmund, März 1994

Universität Dortmund
Fachbereich Informatik

UNI DO

University of Dortmund
Computer Science Department

Forschungsberichte des Lehrstuhls VIII (KI),
Fachbereich Informatik
der Universität Dortmund

ISSN 0943-4135

Anforderungen an:

Universität Dortmund
Fachbereich Informatik
Lehrstuhl VIII
D-44221 Dortmund

Research Reports of the unit no. VIII (AI)
Computer Science Department
of the University of Dortmund

ISSN 0943-4135

Requests to:

University of Dortmund
Fachbereich Informatik
Lehrstuhl VIII
D-44221 Dortmund

e-mail: reports@ls8.informatik.uni-dortmund.de
ftp: kimo.informatik.uni-dortmund.de:pub/Reports

Planen von Aktionen und Reaktionen

Ein einführender Überblick

LS–8 Report 7

Joachim Hertzberg

Dortmund, März 1994

Universität Dortmund
Fachbereich Informatik

Zusammenfassung

Dieser Text gibt einen Überblick über ältere und neuere KI-Methoden zum Planen, also zur Erstellung von Handlungsplänen. Er beleuchtet den Stand der KI-Kunst dazu aus drei miteinander verschränkten Perspektiven: (1) Es gibt eingeschränkte Formen von Handlungsplanung, die gut verstanden sind; der Text stellt die "klassische" Planung kurz vor. (2) Es gibt keine allgemein anwendbaren Verfahren, zielgeleitete Handlungen unter Bedingungen wie Zeitdruck oder Unsicherheit verfügbarer Information zu finden; der Text führt in die entsprechenden Probleme ein und beschreibt einige aktuelle Lösungsansätze. (3) Es gibt Logikkalküle, die Ereignisse und ihre Effekte unter einigen dieser Bedingungen sauber beschreiben bzw. ermitteln können; der Text beschreibt einen solchen Kalkül genauer und skizziert, wie man ihn verwenden kann, um Planer mit klarer Semantik zu bauen, die trotzdem funktionieren.

Dieser Text ist eine leicht überarbeitete Version des Skriptums zu einem Kurs auf der Frühjahrsschule Künstliche Intelligenz (KIFS), Günne, 26. 2. – 5. 3. 1994

Danksagung
Für Vorschläge zur Verbesserung einer Vorversion dieses Textes bedanke ich mich bei Gerhard Kraetzschmar, Guido Lindner, Steffo Weber und Hörern des KIFS-94-Kurses 4.6 „Planen von Aktionen und Reaktionen“.

1 Übersicht

Wie würde ein *wirklich* intelligenter Roboter – also so einer wie in den *Science-fiction*-Filmen – bestimmen, welche Handlung er als nächste auszuführen beginnt?

Auf ihre jeweils eigene Weise befassen sich mehrere Disziplinen mit Aspekten dieser Frage; beispielsweise betrachtet die Kontrolltheorie sie unter technischem Gesichtspunkt etwa bei Handhabungsautomaten, und die Kognitionspsychologie versucht, Modelle des handelnden Menschen aufzustellen, die möglicherweise auf Maschinen übertragbar sind – und sei es nur teilweise oder in Analogie. Die Künstliche Intelligenz (KI) bearbeitet diese Frage ebenfalls, und zwar schon seit den Fünfziger Jahren, also seit es die KI überhaupt gibt.

Alle diese Disziplinen können Antwort-Ausschnitte geben, aber die volle Antwort weiß keiner. Es gibt keine Theorie und kein Computerprogramm, mit denen man beschreiben oder steuern könnte, wie ein Akteur in Alltagssituationen „das Richtige" tut. Andererseits ist die Frage offenbar nicht einfach falsch gestellt, denn es *gibt* ja „Systeme", die das können: uns selber.

Dieser Text gibt einen Abriß der KI-Arbeiten zu der Frage, wie ein Akteur seine jeweils nächste Handlung bestimmt. Das entsprechende KI-Teilgebiet heißt *Planen* (engl. *planning*). Dieser historisch entstandene Name führt allerdings teilweise in die Irre: Eine Zeitlang herrschte die Meinung, ein Akteur bestimme seine nächste Handlung, indem er einen vollständigen Handlungsablauf im vorhinein festlege, eben *plane,* den er dann auszuführen beginne – so etwa wie ein Computerprogramm abgearbeitet wird. Die Grenzen dieses Modells der Handlungssteuerung werden in letzter Zeit deutlich sichtbar, und andere Vorstellungen treten hinzu, die teilweise alternativ sind, teilweise komplementär. Alle solche Arbeiten werde ich ansprechen.

Wenn die KI wie alle anderen Disziplinen die Frage schon nicht abschließend beantworten kann, so sind auch durch ihre Arbeiten der vergangenen knapp 40 Jahre doch wenigstens wichtige Teil- und Unterprobleme klarer geworden. Beispielsweise schälen sich Probleme heraus, die grundsätzlich immer dann bestehen, wenn man modellieren will, was Ereignisse in einer Modellwelt bewirken. Diese Probleme machen auch anderen Gebieten das Leben schwer; um ein modisches Beispiel zu geben: in Systemen zur *virtuellen Realität* (*virtual reality*) treten sie in gleicher Weise auf, wenn die virtuellen Ergebnisse der virtuellen Handlungen des virtuellen Akteurs errechnet und entsprechend dargestellt werden sollen.

Die KI-Arbeiten zum Thema stelle ich in der folgenden Reihenfolge vor:

Abschnitt 2 beschreibt die grundlegenden Probleme beim Erschließen von Handlungseffekten und gibt Strukturmerkmale an, die ein Anwendungsbereich fürs Planen erfüllen oder nicht erfüllen kann, deren Erfülltsein aber Konsequenzen dafür hat, wie „kompliziert" das Planen in diesen Bereichen ist.

Abschnitt 3 gibt eine kurze Einführung ins „klassische" Planen, das durch bestimmte vereinfachende Annahmen über Strukturmerkmale von Anwendungsbereichen und durch die oben genannte Vorstellung ausgezeichnet ist, ein Plan würde zuerst vollständig erstellt und dann wie ein Computerprogramm abgearbeitet.

Abschnitt 4 umreißt Arbeiten (überwiegend neueren Datums), welche die Annahmen aus

Abschnitt 2 verändern, was Konsequenzen für Planungsprozeduren und für das verwendete Modell der Ausführung von Plänen hat.

Abschnitt 5 skizziert das Erschließen von Wandel (engl. *reasoning about change*) durch Aktionskalküle. Auch wenn solche Kalküle in Planern faktisch nicht verwendet werden (und aus pragmatischen Gründen meist gar nicht verwendet werden *können*), bilden sie doch das Material, mit dem man Planer und ihre Ergebnisse formal beschreiben kann und an denen sich diese Ergebnisse messen lassen müssen. Der Abschnitt stellt den Kalkül MW als einen solchen Aktionskalkül ausführlicher vor.

Abschnitt 6 gibt ein Beispiel, wie dieses Sich-messen-Lassen praktisch aussehen könnte: er gibt zunächst eine einfache Prozedur zur Erzeugung von Plänen an, die unvollständige Information über den Anwendungsbereich und kontextabhängige Handlungen mit Effekt-Alternativen behandeln kann. Anschließend definiert er den Begriff „Begrenzt-Korrektheit" bzgl. eines Aktionskalküls und weist nach, daß die angegebene Prozedur korrekt bzgl. MW ist.

Die Beispiel-Anwendungsbereiche in diesem Aufsatz sind durchgängig simpel – etwa die Nilssonsche [Nilsson, 1982] Bauklotzwelt („Blockwelt") oder Modifikationen davon. Der Grund für diese Einfachheit ist *weder,* daß Planungsforscher solche Bereiche für besonders wichtig hielten, *noch,* daß Planung in realen Anwendungsbereichen nicht funktioniert. Die Gründe sind: Didaktik und Faulheit. Ich wollte es Leserinnen, Lesern und mir selber ersparen, erst Anwendungsbereiche wie das Design von Experimenten in der Molekulargenetik, das zielgerichtete Verändern von objektorientierten Bürografiken oder die Steuerung von Prozeßparametern an Plasmadepositionsanlagen einzuführen, bevor ich zum Thema komme. Es ist eine der merkwürdigen Besonderheiten der KI, daß man auf solche Selbstverständlichkeiten anscheinend immer wieder hinweisen muß.

2 Die Problemlandschaft

2.1 Probleme beim Erschließen von Wandel

Weltwissen ist in Raum und Zeit veränderlich. Das liegt zum einen daran, daß die Welt selber sich wandelt und vorhandenes Wissen über sie entsprechend nachgeführt werden muß. Zum anderen ist Wissen im allgemeinen nie vollständig, sicher und korrekt und unterliegt daher ständig der Ergänzung und Korrektur.

Mit dem ersten der beiden Gründe für Veränderlichkeit, und zwar üblicherweise in der Dimension Zeit, befaßt sich das Gebiet *reasoning about change,* auf Deutsch also etwa „Erschließen von Wandel".[1] In diesem Gebiet haben sich im Lauf der Jahre einige grundsätzliche Probleme herauskristallisiert, die zu lösen sind, wenn man – im weitesten Sinne – Kalküle zum Erschließen von Wandel definieren will, oder wenn man Anwendungsprobleme lösen will, in denen Wandel eine Rolle spielt, also beispielsweise wenn man planen will.

[1] Zur Entsprechung englischer Begriffe in diesem Abschnitt 2.1: Es gibt keine anerkannten deutschen Übersetzungen der hier relevanten englischen Begriffe wie *reasoning about change, frame problem, qualification problem.* Um kein Sprachgemisch zu erzeugen, verwende ich deutsche Begriffe, die den englischen in etwa entsprechen. Um die Korrespondenz meiner Begriffe zu den anderswo verwendeten klarzumachen, zeigt Tabelle 1 die in der englischsprachigen Literatur üblichen Bezeichnungen; außerdem sind teilweise Synonyme in Deutsch und Englisch angegeben, die sich in der Literatur ebenfalls finden.

Problem	**Englische Synonyme**	**Deutsche Übersetzungen**
frame pr.	–	Kulissenproblem, *Rahmenproblem,* Trägheitsproblem
persistence pr.	*inertia pr.*	*Beharrungsproblem,* Fortdauerproblem, Persistenzproblem
prediction pr.	*inter-frame pr., extended prediction pr.*	*Vorhersageproblem*
qualification pr.	*initiation pr., intra-frame pr.*	*Qualifikationsproblem,* Voraussetzungsproblem
ramification pr.	–	*Schneeballproblem,* Verzweigungsproblem

Tabelle 1: Zusammenstellung der betrachteten Probleme, englischer Synonymbezeichnungen und deutscher Übersetzungen ihrer Namen. (Die in diesem Text verwendete Übersetzung ist kursiv gesetzt.)

Diese Probleme werden hier vorgestellt. Die Darstellung ist orientiert an [Shoham und McDermott, 1988]; diese Arbeit kann man weiterhin als eine ausführlichere Einführung ins Erschließen von Wandel verwenden. Eine Zusammenstellung der betrachteten Probleme und englischer und deutscher Synonyme ihrer hier verwendeten Bezeichnungen gibt Tabelle 1. Eine Übersicht über diese Probleme findet sich weiterhin in [Sandewall, 1992]; dort sind auch Beispiele zu ihrer Illustration angeführt, und es wird gezeigt, wie unterschiedliche Formalismen zum Erschließen von Wandel mit den Beispielen zurechtkommen (oder auch nicht). Insbesondere betrachtet Sandewall Beispiele für die Probleme, in denen quantitative Zeit eine Rolle spielt. Der Formalismus, den er vorschlägt, bietet zumindest partielle Lösungen für alle diese Probleme an.

Auch wenn Kalküle zum Erschließen von Wandel genauer erst Thema von Abschnitt 5 sind, benötigen wir einen Hintergrund zur Darstellung der Grundprobleme mit allen diesen Kalkülen. Dazu sei deren Urahn schon hier erwähnt: der *Situationskalkül* von McCarthy und Hayes [1969].

Das Prinzip des Situationskalküls ist das folgende. Grundelemente sind *Situationen* und *Operatoren.* Eine Situation ist der Schnappschuß des interessierenden Ausschnitts der Welt zu einem Zeitpunkt, beschrieben durch eine Menge prädikatenlogischer Formeln, die in dieser Situation gelten; ein Operator ist die formale Beschreibung einer Handlung in der Welt.[2] Ein Operator überführt eine gegebene Situation in eine andere Situation: die Nachfolgesituation. Der Situationskalkül ist eine formale Vorschrift, wie man aus einer ge-

[2]Im gesamten Text verwende ich die Begriffe Handlung und Aktion synonym zur Bezeichnung einer Handlung in der Welt. Mit dem Begriff Situation bezeichne ich sowohl einen Ausschnitt der Welt, als auch seine syntaktische Repräsentation; aus dem Kontext ist die Bedeutung jeweils klar.

gebenen Situation s und einem Operator, der in s angewandt wird, die Nachfolgesituation s' errechnen kann.

Die Verwendung von Situation, Operator, Nachfolgesituation und so fort ist die Art, wie im Situationskalkül (und beim klassischen Planen) Zeit betrachtet wird: als das Vorher und Nachher von unterschiedlichen, untereinander widersprüchlichen Situationen beziehungsweise als das Vorher und Nachher von Operatoren.[3] Bei dieser Art der Behandlung von Zeit treten nun einige grundsätzliche Schwierigkeiten auf, die jeder Kalkül zum Erschließen von Wandel behandeln und beispielsweise jedes Planungssystem effizient lösen oder durch entsprechende Maßnahmen umgehen muß.

Vor der Einzeldarstellung der Probleme beim Erschließen von Wandel noch eine Bemerkung, die auf alle diese Probleme zutrifft. Wie immer, wenn man einen Ausschnitt der realen Welt in einem formalen System repräsentiert und auf dieser Repräsentation arbeitet, also beispielsweise Schlüsse zieht, sind die Ergebnisse dieser Schlüsse korrekt nur relativ zum betrachteten Weltausschnitt und zur Richtigkeit der Repräsentation. Wenn man falsche Information repräsentiert hat oder wenn nicht repräsentierte Information relevant gewesen wäre, wird man unsinnige Ergebnisse bekommen. Das ist so in der Logik, in speziellen KI-Wissensrepräsentationsformalismen und in Systemen mit konnektionistisch verteilter Information; und es ist so bei der Repräsentation von Handlungen: es ist also nichts Besonderes.

Wenn man beispielsweise einen Weltausschnitt in einer Menge von Formeln F einer logischen Sprache repräsentiert hat, kann man alle Information über diesen Weltausschnitt bekommen, indem man die vollständige inferentielle Hülle von F berechnet. Es kann jedoch sinnvoll sein und ist hier relevant, bewußt und gezielt nicht alle vorhandene Information zu benutzen, weil es beispielsweise zu teuer ist, die inferentielle Hülle von F zu bilden. Dann ist ein Schluß, den man gezogen hat, möglicherweise falsch, obwohl man „es hätte besser wissen können". Wenn ich im folgenden von der Inkorrektheit oder Unvollständigkeit von Information oder Ableitungen spreche, meine ich diese gezielte Inkorrektheit, nicht die triviale Inkorrektheit, die jeder Repräsentation anhaftet. Shoham und McDermott bringen die Sache auf den Punkt: Die nachfolgend vorgestellten Probleme beim Erschließen von Wandel sind

> problems that arise from the conflicting desires to reason both rigorously and efficiently about the future. [Shoham und McDermott, 1988, S. 50]

2.1.1 Das Qualifikationsproblem

Das Qualifikationsproblem geht zurück auf [McCarthy und Hayes, 1969]. Es ist

> das Problem, die Zukunft vorherzusagen, ohne alles zu berücksichtigen, was man über die Vergangenheit weiß.

Entgegen einem verbreiteten Mißverständnis hat das Qualifikationsproblem nichts mit der trivialen Tatsache zu tun, daß es unmöglich ist, alle Vorbedingungen anzugeben, die *grundsätzlich* gelten müssen, damit eine Handlung ausgeführt werden kann: Das ist für alle Weltausschnitte interessanter Vielfältigkeit zumindest praktisch unmöglich. (Inwieweit

[3] Tatsächlich betrachtet man im Situationskalkül beliebige *Ereignisse,* nicht nur Handlungen. Das soll hier nicht interessieren.

es aus tieferliegenden, eher philosophischen Gründen unmöglich ist, will ich gar nicht erörtern.) Wenn also in der Literatur Ansätze zur Lösung des Qualifikationsproblems behandelt werden, ist das immer so zu verstehen, daß versucht wird, Vorhersagen über zukünftige Verhältnisse zu treffen und dabei nur *Teile* des praktisch vorhandenen Wissens berücksichtigen zu müssen.

Die typische Gestalt des Qualifikationsproblem beim Planen ist das Problem, die Anwendbarkeit eines Operators zu beurteilen. Im allgemeinen verbietet es sich eben, alle Merkmale in die Vorbedingungen eines Operators aufzunehmen und folglich zur Laufzeit des Planers zu berücksichtigen, die *tatsächlich* gelten müssen, damit er ausführbar ist. Voraussetzungen, die normalerweise als gültig anzusehen sind, läßt man der Effizienz halber gerne weg, selbst wenn sie in der verwendeten Modellierung des Anwendungsbereichs vorhanden sind. Gilt eine solche Bedingung im Einzelfall aber dann doch nicht, resultiert aus der formalen Anwendung des Operators im allgemeinen eine Situation, die mit der sonstigen Bereichsbeschreibung inkonsistent ist. Ginsberg [Ginsberg, 1993, S. 281] formuliert diesen planungsbezogenen Aspekt des Qualifikationsproblem so:

> An action may fail because its success would involve the violation of a domain constraint.

Das Standard-Lehrbeispiel für das Qualifikationsproblem ist das Problem der Kartoffel im Auspuff.[4] Das Beispiel dreht sich darum, daß man als Vorbedingungen der Aktion, ein Auto anzulassen, normalerweise betrachten würde, daß der Zündschlüssel steckt, die Batterie geladen ist, Benzin im Tank ist und dergleichen. Weiterhin ist grundsätzlich auch eine Vorbedingung für diese Aktion, daß *keine* Kartoffel im Auspuff steckt. (Und auch keine Möhre und keine Kiwi. Es sei denn, die Kartoffel, Möhre oder Kiwi hat senkrecht zur Auspuffrohrachse gemessen einen kleineren Durchmesser als das Auspuffrohr.) Gebe es also eine entsprechende Regel, oder ein *domain constraint* in Ginsbergs Terminologie, in der Bereichsdarstellung, die besagt, daß Autos mit Kartoffeln oder Kiwis im Auspuff nicht laufen können. Aber selbst wo diese Information vorhanden ist, möchte man nicht immer gezwungen sein, sie zu beachten, wenn die Anwendbarkeit der Aktion Anlassen beurteilt werden soll, einfach weil sie normalerweise gilt.

Typische Vorschläge zur Lösung des Qualifikationsproblems [Shoham, 1988; Ginsberg und Smith, 1988b] verwenden nichtmonotone Logiken, um die als *normalerweise gültig* verstandenen Vorbedingungen als *tatsächlich gültig* anzunehmen, diese Annahme bei widersprüchlicher Information aber zurücknehmen zu können.

2.1.2 Das Vorhersageproblem

Das Vorhersageproblem ist

> das Problem, vorherzusagen, welche Fakten nach einer Folge von Operatoren wahr sein werden.

Manche Autoren, wie beispielsweise Shoham und McDermott [1988] unterscheiden zwischen dem Vorhersageproblem und dem *erweiterten* Vorhersageproblem (*extended prediction problem*), wobei das erweiterte Vorhersageproblem darin liegt, über eine beliebige

[4] Wie Sandewall [1992] schreibt, liegt die Urheberschaft dieses Beispiels im Dunkeln.

Folge von Operatoren hinweg vorherzusagen, und das „normale" lediglich über einen als atomar angesehenes Operator. Ich folge dieser Unterscheidung hier nicht.

Die spezielle Schwierigkeit, das Vorhersageproblem zu lösen, liegt darin, daß Handlungen im allgemeinen nicht „kontextfrei" ablaufen, sondern daß vorher oder überlappend ablaufende Handlungen das Ergebnis einer einzelnen Handlung beeinflussen. Wiederum möchte man aus Gründen der Berechnungskomplexität nicht alle bekannten anderen Handlungen ansehen müssen, um den Effekt einer Einzelhandlung zu berechnen – obwohl man es theoretisch natürlich könnte –, und trotzdem ein „vernünftiges" Ergebnis bekommen. Das Problem wird also wieder dadurch zum Problem, daß man an sich vorhandene Information nicht vollständig auswerten will.

Als Beispiel für das Vorhersageproblem geben Shoham und McDermott [1988] das Problem an, den Lauf einer Billardkugel auf einem Billardtisch vorherzusagen, wobei sie voraussetzen, daß die gesamte Laufstrecke zusammengesetzt ist aus kleinen, atomaren Strecken, die sie als Elementarereignisse ansehen. Andere bekannte Typen von Ereignissen im Problembereich sind die Kollision einer Kugel mit der Bande, das Laufen anderer Kugeln, das urplötzliche Herabstoßen von Adlern auf den Billardtisch und andere. Pragmatisch gesehen sollte man eine Vorhersage über die Position, Rollrichtung und Geschwindigkeit einer Kugel nach einer bekannten Folge von Ereignissen daran orientieren, was „um sie herum" vorgeht, also berücksichtigen, ob in der aktuellen Rollrichtung eine Bande nahe ist, ob andere Kugeln die Bahn kritisch kreuzen, ob man bereits einen Adler hat schreien hören und Ähnliches. Die Schwierigkeit ist, diese pragmatische Sicht praktisch umzusetzen.

Man beachte, daß das Vorhersageproblem vom Qualifikationsproblem in dem Sinne unabhängig ist, daß man es auch dann immer noch hätte, wenn das Qualifikationsproblem gelöst wäre. Andersherum hängen alle folgenden Probleme mit dem Vorhersageproblem eng zusammen; sie sind als Unterprobleme des Vorhersageproblem zu verstehen, die sich darin unterscheiden, ob man auf die *Änderung* oder die *Beharrung* von Fakten in späteren Zuständen fokussiert.

2.1.3 Das Schneeballproblem

Das Konzept Schneeballproblem geht zurück auf Finger [1987]; die Bedeutung dort ist allerdings etwas anders als die, in welcher der Begriff später in der Planungsliteratur verwendet wird. (Finger betrachtet *ramifications,* also Schneeballeffekte, von *Zielen.*) Das Schneeballproblem ist ein Unterproblem des Vorhersageproblems:

> das Problem, vorherzusagen, welche weiteren Fakten wahr werden, wenn in einer Situation S ein Faktum F wahr *wird.*

Ein anderes Unterproblem des Vorhersageproblems behandelt Abschnitt 2.1.4: die Frage, ob ein Faktum, das bereits wahr ist, nach einer Folge von Operatoren auch wahr *bleibt.*

Ein Beispiel für das Schneeballproblem tritt in der Blockwelt auf, wenn wir voraussetzen, es sei ein Prädikat Über als transitive Hülle des üblichen Blockweltprädikats Auf definiert. Nehmen wir an, wir wollen den Operator $\mathbf{STAPLE}(x, y, s)$ definieren, der die Handlung beschreibt, in Situation s Block x auf Block y zu setzen. Eine gleichsam *lokale* Nachbedingung dieses Operators ist $\mathsf{Auf}(x, y, s')$: in der resultierenden Situation s' steht x auf y. Doch was, wenn y bereits in s seinerseits auf einem z steht? Dann (und nur dann) ist $\mathsf{Über}(x, z, s')$ eine weitere Nachbedingung; desgleichen $\mathsf{Über}(x, u, s')$ für alle Blöcke u, für die $\mathsf{Über}(y, u, s)$ gilt. Weiterhin möge ein Prädikat $\mathsf{Turm}(s)$ definiert sein, das wahr ist, wenn es in s einen Stapel von mindestens vier Blöcken Höhe gibt; dieses Prädikat kann ebenfalls in s' wahr werden – oder auch nicht. Man kann sich denken, daß jede Aktion abhängig vom Kontext eine Kette von Schneeballeffekten haben kann, aber eben nicht haben muß.

Das Problem beim Schneeballproblem ist also wieder „nur" Komplexität. Wenn die direkte, kontextunabhängige Wirkung einer Aktion in allen interessierenden Teilen vollständig beschrieben ist und wenn man aus einer Situation, in der ein Operator ausgeführt wird, und dessen direkter Nachbedingung den „Kern" der Beschreibung der Nachfolgesituation bestimmen kann, dann müßte man lediglich die volle inferentielle Hülle dieses Situationskerns bilden. Im allgemeinen ist das aber zu aufwendig, so daß man versucht, mit lediglich den „wichtigen" kontextabhängigen Effekten weiterzuarbeiten.

Es gibt keine ausdrücklichen Ansätze, das Schneeballproblem insgesamt zu behandeln; eher theoretische Ansätze zur Formalisierung von Aktionen können es sich leisten, eben *doch* die Berechnung der inferentiellen Hülle vorauszusetzen. Einen Formalismus zur Behandlung kontextabhängiger Effekte beim Planen entwickelt [Pednault, 1988]. Ich werde im Abschnitt 5 auf das Problem zurückkommen.

2.1.4 Das Rahmenproblem und das Beharrungsproblem

Das Rahmenproblem ist neben dem Qualifikationsproblem dasjenige Problem, das in der Literatur am ausführlichsten besprochen worden ist, und es ist neben diesem zuerst als ganz grundlegendes Problem im Erschließen von Wandel erkannt worden [McCarthy und Hayes, 1969]. Allerdings mag die häufige Erwähnung des Begriffes Rahmenproblem in der Literatur unter anderem daran liegen, daß unter diesen Begriff mitunter andere der hier besprochenen Probleme subsumiert werden, ja, daß der Begriff Rahmenproblem geradezu als Synonym für die Probleme im Erschließen von Wandel insgesamt verwandt wird. Eine Quelle mit Texten über das Rahmenproblem in diesem weiteren Sinn ist [Pylyshyn, 1987].

Das Rahmenproblem und das Beharrungsproblem sind zwei eng verwandte Probleme. Das Beharrungsproblem ist

> das Problem, vorherzusagen, welche Fakten über eine Folge von Operatoren gültig bleiben.

Damit ist das Beharrungsproblem also ein Spezialfall des Vorhersageproblems.

Beim Planen werden wir in Abschnitt 3 ein Beharrungsproblem spezieller Form bekommen: das Problem, Konflikte in nichtlinearen Plänen zu erkennen. Intuitiv ist das Problem hier, daß eine Aktion eine Vorbedingung erfordert, die von einer anderen Aktion erzeugt wird, jedoch zerstört eine dritte Aktion diese Bedingung möglicherweise wieder; dadurch wird ein Plan möglicherweise unausführbar. Demnach liegt also ein Beispiel für ein Beharrungsproblem vor: man muß erkennen, ob eine einmal erzeugte Bedingung bis hin zu der Aktion gültig bleibt, die sie als Vorbedingung erfordert. Wie in [Hertzberg und Horz, 1989] klargemacht wird, kann die Lösung selbst dieses Spezialfalles des allgemeinen Beharrungsproblems im allgemeinen beliebig komplex werden.

Das Rahmenproblem ist gleichsam ein „lokales" Beharrungsproblem. In der hier verwendeten engen, technischen Bedeutung bezeichnet es

> das Problem, in der Nachbedingung eines Operators anzugeben, was alles sich durch ihn *nicht* ändert.

Dieses Problem tritt nicht in allen denkbaren Formalismen auf, Handlungen zu beschreiben, jedoch in einem, der klassischerweise verwendet worden ist: Prädikatenlogik erster Stufe.

Als Beispiel betrachten wir wieder den Blockwelt-Operator **STAPLE**. Etwas genauer als eben beschrieben, soll er folgendes leisten: Er ist anwendbar, wenn Block x in der Greifhand gehalten wird und y frei zugänglich ist; als Ergebnis gelten die beiden Vorbedingungen nicht mehr, die Hand hält nichts mehr, x ist frei zugänglich und steht auf y. Unter Verwendung intuitiv entsprechender Prädikate beschreibt dann das folgende Axiom den Operator (s' sei die Abkürzung für den Situationsterm $\mathbf{STAPLE}(x, y, s)$; alle anderen Variablen seien allquantifiziert):

$$\mathsf{Hand}(x,s) \wedge \mathsf{Frei}(y,s) \rightarrow \neg\mathsf{Hand}(x,s') \wedge \neg\mathsf{Frei}(y,s') \wedge \mathsf{Hand}(\textsc{nil},s') \wedge \mathsf{Frei}(x,s') \wedge \mathsf{Auf}(x,y,s')$$

Sei nun die Situation s_0 gegeben durch die Formel

$$\mathsf{Hand}(\mathrm{A}, \mathrm{s}_0) \wedge \mathsf{Frei}(\mathrm{B}, \mathrm{s}_0) \wedge \mathsf{Tisch}(\mathrm{B}, \mathrm{s}_0)$$

für die Blöcke A und B. (Tisch bedeute, der entsprechende Block steht auf dem Tisch.) Offenbar gelten in s_0 die Vorbedingungen von **STAPLE**. Folglich kann man ihn anwenden und erhält

$$\neg\mathsf{Hand}(\mathrm{A}, \mathrm{s}_1) \wedge \neg\mathsf{Frei}(\mathrm{B}, \mathrm{s}_1) \wedge \mathsf{Hand}(\textsc{nil}, \mathrm{s}_1) \wedge \mathsf{Frei}(\mathrm{A}, \mathrm{s}_1) \wedge \mathsf{Auf}(\mathrm{A}, \mathrm{B}, \mathrm{s}_1)$$

Die Intuition sagt, daß B in s_1 natürlich immer noch auf dem Tisch steht. Wie können wir das ableiten?

Gar nicht! Man müßte zusätzlich zur vorhandenen Beschreibung von **STAPLE** noch sagen, daß sich an der Eigenschaft Tisch aller Blöcke nichts ändert, also ein weiteres Axiom zur Beschreibung von **STAPLE** formulieren:

$$\forall x, y, z, s. [\mathsf{Tisch}(x, s) \rightarrow \mathsf{Tisch}(x, \mathbf{STAPLE}(y, z, s))]$$

Ein solches Axiom hieße analog zum Rahmenproblem ein *Rahmenaxiom.*

Wenn es jedoch im Problembereich weitere Prädikate gäbe, die durch **STAPLE** ebenfalls nicht geändert würden, müßte man für alle diese Prädikate ebenfalls separate Rahmenaxiome angeben. Dadurch würde sich die Beschreibung eines jeden Operators bis zu beliebiger Uneleganz aufblähen. Die *technische* Lesart des Rahmenproblem lautet daher: Wie gibt man auf *ökonomische* Weise an, was alles ein Ereignis *nicht* in seiner Nachbedingung ändert.

Eine Lösung des Rahmenproblem wurde für Planungssysteme schon früh im System STRIPS [Fikes und Nilsson, 1971] gefunden: man ging praktisch von einer direkten Formalisierung der Aktionseffekte ab, modellierte damit logisch gesehen aber so etwas wie ein implizites Axiom zweiter Stufe. („Alles, was nicht geändert wird, bleibt gleich.") Später hat Lifschitz [1987] STRIPS-Plänen eine Semantik unter Verwendung von Logik erster Stufe gegeben, die allerdings nicht trivial ist; ich werde in Abschnitt 5 darauf kommen.

Eine auf den ersten Blick naheliegende, andere Lösung des Rahmenproblems besteht darin, Default-Logik zu verwenden, um auszudrücken, daß sich Fakten durch Ereignisse *normalerweise* nicht ändern, wobei der Normalfall dadurch außer Kraft gesetzt wird, daß ein Faktum in der Nachbedingung eines Ereignisses ausdrücklich als geändert angegeben ist. Wie man sieht, führt diese naheliegende Idee ohne weiteres jedoch in eine Sackgasse, wie man an dem sogenannten *Yale shooting problem* sieht. Dieses Problem wird im folgenden Exkurs beschrieben; wer diesen Exkurs auslassen will, lese bei 2.2 weiter.

Exkurs: Das *Yale shooting problem*

Das *Yale shooting problem* liegt auf einer anderen Ebene als die vorher angesprochenen Probleme. Es ist eigentlich kein Problem im Erschließen von Wandel im allgemeinen, sondern ein Beispiel für ein Problem mit einem speziellen Versuch der Lösung des Rahmenproblems. Da es seit seiner Entdeckung [Hanks und McDermott, 1986] in den letzten Jahren intensiv bearbeitet worden ist, schließe ich es in dieser Übersicht ein.

Das Problem besteht darin, wie die Beharrung von Fakten über die Anwendung von Operatoren erzielt wird, nämlich so: Mit Mitteln der Default-Logik wird formalisiert, daß alle Fakten, die nicht in der Nachbedingung einer Aktion geändert werden, als gleichbleibend erschlossen werden. Das Problem ist unabhängig von dem verwendeten nichtmonotonen Logik-Formalismus; ich verwende im folgenden Beispiel eher intuitiv eine Logik, die an Reiters [1980] Default-Logik angelehnt ist.

Das Beispiel ist das folgende. Ein Individuum namens Fred ist entweder Alive oder $\neg\mathsf{Alive}$.[5] Weiterhin gibt es ein Gewehr, für das Loaded oder $\neg\mathsf{Loaded}$ gelten kann. Es gibt den Operator **LOAD**, das Gewehr zu laden, den Operator **SHOOT**, auf Truthahn Fred zu schießen, und den Operator **WAIT**, gar nichts zu tun. Die Vor- und Nachbedingungen dieser Operatoren sind die intuitiv vermuteten; allerdings wird vorausgesetzt, daß **SHOOT** immer trifft, mit dem für den Truthahn fatalen Ergebnis $\neg\mathsf{Alive}$.

Wenn in der Anfangssituation s_0 Alive gilt, dann haben wir die folgenden Axiome, wobei die Variable s überall allquantifiziert sei:

$$\begin{array}{l}\mathsf{Alive}(\mathrm{s}_0)\\ \mathsf{Loaded}(\mathbf{LOAD}(s))\\ \mathsf{Loaded}(s) \rightarrow \neg\mathsf{Alive}(\mathbf{SHOOT}(s))\end{array}$$

Für den Default-Schluß darauf, daß sich alle in Operator-Nachbedingungen nicht erwähnten Fakten nicht ändern, gebe es das folgende Schema einer Default-Regel, wobei Instanzen dieser Regel dadurch entstehen, daß man in allen möglichen Kombinationen p durch eines der vorhandenen Prädikate und a durch eine der vorhandenen Operatoren ersetzt:

$$\frac{p(s) :\ \mathrm{M}\, p(a(s))}{p(a(s))}$$

zu lesen als: wenn $p(s)$ gilt und $p(a(s))$ konsistent mit aller weiteren Information ist, dann leite $p(a(s))$ ab. Eine Instanz dieser Regel nenne ich auch *frame default*.

Nehmen wir nun an, daß der Reihe nach die Operatoren **LOAD**, **WAIT** und **SHOOT** ausgeführt werden, die von s_0 aus am Ende zu s_3 führen; schematisch also:

$$\mathrm{s}_0 \underset{\mathbf{LOAD}}{\Longrightarrow} \mathrm{s}_1 \underset{\mathbf{WAIT}}{\Longrightarrow} \mathrm{s}_2 \underset{\mathbf{SHOOT}}{\Longrightarrow} \mathrm{s}_3$$

[5] Nach Sandewall [1992] ist Fred ein Truthahn. Ich möchte dieser Interpretation des Beispiels folgen.

	1		2		3	
s_0	Alive		Alive		Alive	
s_1	Alive	Loaded	¬Alive	Loaded	Alive	Loaded
s_2	Alive	Loaded	Alive	Loaded	Alive	¬Loaded
s_3	¬Alive	Loaded	¬Alive	Loaded	Alive	¬Loaded

Tabelle 2: Drei mögliche Situationsfolgen, die sich mit den vorhandenen Axiomen und *frame defaults* rechtfertigen lassen.

Intuitiv ist klar, was passieren soll: in s_1 wird das Gewehr geladen sein, von dem man anfangs nicht wußte, ob es geladen ist oder nicht; durch das Warten soll sich nichts ändern; und nach dem Schießen wird der Truthahn tot sein. Im übrigen soll immer alles gleich bleiben.

Dieser Gang durch die Situationen s_0 bis s_3 ist aus den Axiomen mit den angegebenen *frame defaults* tatsächlich ableitbar. Das *Yale shooting problem* entsteht nun dadurch, daß noch *weitere* Situationsfolgen ableitbar sind, die zum Teil nach *formalen* Kriterien nicht besser oder schlechter sind als die intuitiv erwartete.

In Tabelle 2 sind drei ableitbare Situationsfolgen angegeben. Dabei zeigt die mit 1 bezeichnete Spalte den beschriebenen, intuitiven Gang der Handlung. Spalte 2 ist, intuitiv betrachtet, sehr merkwürdig. Beispielsweise ist nach **LOAD** in s_1 Fred auf einmal ¬Alive. Dieser Schluß ist zwar – wie die Intuition es nahelegt – durch die verwendeten Axiome und *frame defaults* nicht *erzwungen,* aber diese sagen auch nichts dagegen, ein Faktum wie ¬Alive in s_1 einfach „per Hand" einzufügen. Folglich verhindern sie ebenfalls nicht, daß in s_2 Fred wieder Alive ist, um dann in s_3 wie vermutet wieder ¬Alive zu sein.

Im Verlauf dieses Gangs durch die Situationen wird also dreimal ein *frame default* überschrieben, der normalerweise ein Faktum in die Nachfolgesituation hinüberretten würde: bei jedem Wechsel von Alive in ¬Alive und umgekehrt. Einer dieser Wechsel, von s_2 nach s_3, wird durch ein Operator-Axiom bewirkt, die anderen sind sozusagen zufällig. In der intuitiv richtigen Situationenfolge 1 in Tabelle 2 wird ein *frame default* hingegen nur einmal überschrieben, und zwar durch eine Operator-Nachbedingung. Man könnte nun sagen, ein gutes syntaktisches Kriterium zur Unterscheidung von intuitiven und nicht intuitiven Situationsfolgen sei die Anzahl der Überschreibungen eines *frame default*.

Dagegen spricht die Situationsfolge in Spalte 3. Auch hier wird nur einmal ein *frame default* überschrieben, und zwar beim Übergang von s_1 nach s_2: das Gewehr wird während des Wartens auf wundersame Weise entladen. Dennoch würde man diese Situationsfolge wohl nicht als genauso intuitiv akzeptieren wie die in Spalte 1.

Es gibt eine ganze Reihe von Vorschlägen, das *Yale shooting problem* zu lösen, ohne davon abzugehen, das Rahmenproblem durch die Verwendung irgendeiner Variante von *frame defaults* zu lösen. Eine genauere Übersicht über diese Ansätze gibt wiederum [Sandewall, 1992].

2.2 Was ist Planen?

Viele der Computerprogramme, Algorithmen oder Methoden zum Planen, die ich später bespreche, umgehen die genannten Probleme eher, statt sie wirklich zu behandeln – was pragmatisch verständlich ist. Die Strategie, vereinfachende Annahmen über die Anwendungsbereiche zu machen, unterliegt in irgendeiner Weise praktisch allen Arbeiten zum Planen. *Das* Planungsproblem *im allgemeinen* ist mit realistischem Aufwand nicht lösbar. Allerdings sollte man sich die vereinfachenden Annahmen klarmachen, die ein betrachtetes Planungssystem oder eine Planungsprozedur voraussetzen. Das geschieht in der Literatur zu selten ausdrücklich.

Nach Ginsberg [1993] stellen sich beim Problemlösen im allgemeinen und beim Planen im besonderen zwei große Problemkreise: *Repräsentation* des Anwendungsbereichs und *Steuerung* der heuristischen Suche durch den Suchraum, der durch die Repräsentation implizit vorgegeben ist. Beide Problemkreise kann man vereinfachen durch einschränkende Annahmen über die Planungsprobleme, die man behandeln können möchte: Repräsentation vereinfacht man durch Annahmen über die Struktur des Anwendungsbereichs, und Steuerung durch Anforderungen an zu findende Pläne oder an den Planer. Ich will einige solche Vereinfachungen aufzählen, die man im Planen machen kann – oder eben nicht macht.[6] Fast unnötig zu betonen, daß nicht nur das Planen, sondern natürlich auch Verfahren zum Erschließen von Wandel sich ändern müssen, wenn es die Charakteristik des Anwendungsbereichs tut. Ich werde darauf in Abschnitt 5 zurückkommen.

Die im folgenden aufgeführten Eigenschaften kann man in Planungs-Anwendungsbereichen vorfinden:[7]

Zeitmodell: Situationen oder Intervalle. Sollen komplette „Schnappschüsse" der Welt betrachtet werden, bei denen Ereignisse oder Aktionen vom einen zum anderen überführen? Oder kommt es auf Zeitintervalle und deren Verhältnis zueinander an, innerhalb derer Fakten gelten?

Zeitmodell: Qualitativ oder quantitativ. Sind numerische Zeitangaben wesentlich, wie beispielsweise Uhrzeiten oder Planausführungszeiten? Oder ist man nur am relativen Verhältnis von Aktionen, Situationen oder Gültigkeitsintervallen interessiert?

Bereichsinformation bei Planung: Vollständig oder unvollständig. Ist die Situation, in der die Planausführung startet, eindeutig bzw. unter Verwendung von Bereichsaxiomen vervollständigbar? Oder gibt es Alternativen?

Bereichsinformation bei Ausführung: Vollständig oder unvollständig. Ist bei Planausführung alle relevante Information im Anwendungsbereich jederzeit vorhanden oder vernachlässigbar einfach erhältlich? Oder sind Fakten grundsätzlich unbekannt oder nur mit wesentlichem Zeit- oder Kostenaufwand sicher erfahrbar?

[6] Die Aufzählung ist leicht erweitert aus [Hertzberg, 1993] übernommen, wo die Vereinfachungen zusätzlich ein wenig diskutiert werden.

[7] Dazu eine Bemerkung: Diese Liste ist weder sicher vollständig, noch sind die Eigenschaften orthogonal (es kann also sein, daß einige nur in Kombination mit anderen auftreten können), noch sind sie „epistemologisch adäquat" (beispielsweise kann Zeit im allgemeinen wesentlich tiefergehend modelliert werden als hier angesprochen). Die Liste ist pragmatik-orientiert in dem Sinne, daß sie ein brauchbares Raster ergibt, um KI-Arbeiten zum Planen einzusortieren.

Aktionen: Kontextfrei oder kontextabhängig. Haben gleiche Aktionen unter allen Umständen gleiche Wirkungen, zumindest im betrachteten Abstraktionsgrad? Oder hängen die Effekte im Detail von der Anwendungssituation ab? Gibt es nebenläufig ausgeführte Aktionen, deren Effekte sich von der Summe der Effekte der Einzelaktionen unterscheiden?

Aktionseffekte: Deterministisch oder nicht-deterministisch. Sind alle Effekte von Aktionen in allen möglichen Kontexten ihrer Anwendung sicher vorhersagbar? Oder treten unterschiedliche Effekte auf, deren Eintrittsbedingungen im verwendeten Abstraktionsgrad der Bereichsdarstellung nicht unterschieden werden können?

Ziele: Direkt erreichbar, über- oder unterspezifiziert. Sind die Planziele konsistent? Gibt es Aktionen, die sie erfüllen können? Oder sind sie widersprüchlich? Müssen sie als Teil des Planens erst präzisiert oder operationalisiert werden?

Anwendungsbereich bei Planung: Dynamisch oder statisch. Kann sich die Problemsituation während der Planung ändern? Oder ist die Welt derweil „eingefroren“, bzw. die Rate von Veränderung vernachlässigbar?

Anwendungsbereich bei Ausführung: Dynamisch oder statisch. Können zur Zeit der Plananwendung Ereignisse auftreten, welche die Planausführung nicht kontrolliert? Oder ereignen sich genau die Aktionen, die der Plan vorschreibt, und ereignen sie sich genau so wie repräsentiert?

Planerzahl: Einer oder mehrere. Gibt es genau einen Planer, der an der Veränderung des Anwendungsbereichs plant? Oder gibt es andere, mit denen er zusammen planen kann? Wenn ja: Hat der Planer ausreichende Information über die anderen? Sind sie kooperativ, neutral oder gar feindselig?

Fallwissen: Vorhanden oder nicht vorhanden. Sind fürs Planen Pläne oder Planungsprotokolle aus dem Anwendungsbereich als Informationsquelle verfügbar? Oder muß beim Planen bei Null angefangen werden?

Das „klassische“ Planen, das lange Zeit die Literatur dominiert hat und das in Abschnitt 3 genauer beschrieben wird, war beispielsweise etwa durch die folgenden Voraussetzungen gekennzeichnet:[8] Situationsbasiert bei qualitativer Zeit; vollständige Bereichsinformation bei Planung und Ausführung; kontextfreie, deterministische Aktionen; konsistente, operationalisierte Ziele; Anwendungsbereich bei Planung und Ausführung statisch; ein einziger Planer; kein Fallwissen.

Anforderungen an Pläne oder Planer könnten etwa folgendermaßen aussehen:

Pläne: Korrekt oder nicht. Soll ein fertiger Plan korrekt sein bzgl. der Bereichsbeschreibung, also z.B. nur garantiert anwendbare Operatoren enthalten? Oder darf er gemessen an der Bereichsbeschreibung Fehler enthalten?

[8]Ich schreibe, es war *etwa* durch diese Voraussetzungen gekennzeichnet, weil tatsächlich nie völlige Einmütigkeit darüber geherrscht haben dürfte, zumal die Voraussetzungen nur in sehr wenigen Arbeiten expliziert worden sind.

Pläne: Vollständig oder nicht. Soll ein fertiger Plan vollständig sein bzgl. der Bereichsbeschreibung, also z.B. alles berücksichtigen, was bekanntermaßen bei seiner Ausführung passieren kann? Oder darf er gemessen an der Bereichsbeschreibung Lücken enthalten?

Pläne: Optimal oder nicht. Soll ein fertiger Plan optimal sein bzgl. eines anzugebenden Maßes, also z.B. Länge des Plans? Oder sind auch andere Pläne akzeptabel? Soll es einen Schwellwert geben, oberhalb dessen ein Plan akzeptabel ist? Hängt die Qualität eines Plans von den Ressourcen (z.B. Rechenzeit) ab, die der Planer zu seiner Erstellung verbraucht hat?

Planer: Korrekt oder nicht. Soll der Planer korrekt sein bzgl. eines zu spezifizierenden Planer-Korrektheitsbegriffs, also z.B. nur optimale, korrekte und vollständige Pläne abliefern? Oder darf er – möglicherweise auch nur am Anfang der Bearbeitung eines Problems – inkorrekt sein?

Planer: Vollständig oder nicht. Soll der Planer vollständig sein bzgl. eines zu spezifizierenden Planer-Vollständigkeitsbegriffs, also z.B. alle korrekten und vollständigen Pläne abliefern? Oder reicht Teil-Vollständigkeit aus?

Planer: Ressourcenabhängig oder nicht. Soll das Verhalten des Planers, insbesondere ein erzeugter Plan, in Abhängigkeit der Ressourcen des Planers beurteilt werden, also z.B. abhängig von der verbrauchten Rechenzeit? Soll der Planer unter engen Realzeitanforderungen arbeiten können? Oder spielen Rechenressourcen keine Rolle?

Planer: Fallbasiert oder nicht. Soll der Planer vorhandenes Fallwissen verwenden können, also z.B. eine Bibliothek von Problemen und zugehörigen Lösungsplänen? Soll er Pläne, die er erstellt, oder Protokolle ihrer Erstellung einer vorhandenen Fall-Basis geeignet indiziert hinzufügen können? Oder soll er ohne Rücksicht auf etwa vorhandene Information arbeiten?

Unter dem Sammelbegriff „klassische Planer“ werden beispielsweise meist solche Planer verstanden, die korrekt sind bezüglich des Merkmals, korrekte und vollständige Pläne zu erzeugen; oft wird Optimalität von Plänen bzgl. Zahl oder Kostensumme der Operatoren verlangt; Ressourcenabhängigkeit und Fallbasierung ist ausgeschlossen; Planer-Vollständigkeit ist nicht gefordert.

Auch wenn möglicherweise nicht alle denkbaren Kombinationen von Merkmals-Ausprägungen von Planern, Plänen und Planungs-Anwendungsbereichen sinnvoll sind, ist offensichtlich, daß es sehr viele davon geben kann. Das ist deshalb wichtig zu verstehen, weil unterschiedliche Voraussetzungen an den Bereich und unterschiedliche Anforderungen an Pläne erzwingen, unterschiedliche Algorithmen oder Unterprozeduren zum Planen verwenden. Geht man beispielsweise von situationsbasierter qualitativer Zeit über zu intervallbasierter quantitativer, muß man nicht nur die numerischen Zeitangaben über Intervallgrenzen neu behandeln, sondern es kommt auch hinzu, daß Handlungen nun überlappend oder gleichzeitig ausgeführt werden können, statt wie vorher streng sequentiell; dadurch wird es nötig, zu beschreiben und im Planer zu berücksichtigen, wie sich überlappend ausgeführte Handlungen beeinflussen, und der Planer bekommt zum Beispiel mehr

Freiheit, Konflikte zwischen Operatoren aufzulösen, also zu beseitigen, daß zwei Handlungen im Plan „sich stören“. Weiterhin ist die Ausführungsdauer eines Plans nicht mehr nur von der Anzahl der darin enthaltenen Operatoren abhängig, was wichtig ist für die Definition von Maßen für die Plangüte. Und viele Kleinigkeiten mehr ändern sich – dabei hat der Schritt von einem zum anderen Zeitmodell ja nur zwei der Eigenschaften von Anwendungsbereichen modifiziert!

Was Planen ist, kann man sinnvoll und präzise also nur relativ zu den Eigenschaften des Anwendungsbereichs und den Anforderungen an Planer und Pläne sagen. Um die Richtung anzudeuten, will ich hier dennoch den Zentralbegriff im Planen, nämlich *Plan,* ein wenig eingrenzen auf eine Weise, die die meisten KI-Arbeiten zum Planen umfaßt.

Begriffsbestimmung 1 (Plan) *Ein Plan ist eine Struktur, welche Repräsentationen von Handlungen und Zielen enthält und dazu dient, über die Wirkung zukünftiger Handlungen zu räsonieren und die zielgerichtete Ausführung von Handlungen zu beeinflussen.*

Diese Begriffsbestimmung ist bewußt sehr weit gefaßt, wenn sie auch bereits einiges ausschließt: Übliche imperative Programme fallen beispielsweise nicht darunter, da in ihnen die Ziele nicht repräsentiert sind, die mit ihnen insgesamt oder mit einzelnen Anweisungen verfolgt werden. Technische Definitionen des Begriffs Plan müssen die in dieser Begriffsbestimmung vage gehaltenen Begriffe wie *Struktur, Repräsentation, Wirkung, räsonieren* oder *beeinflussen* präzisieren, wobei das in unterschiedlichen Versionen des Planens unterschiedlich aussehen wird.

Eine solche Version des Planens betrachten wir nun genauer: das klassische Planen.

3 Schnellkurs in „klassischer“ Planung

In Abschnitt 2.2 habe ich das klassische Planen bereits charakterisiert durch die Einschränkungen, die es bezüglich der Eigenschaften von Planungsbereichen und Anforderungen an Pläne und Planer macht. Der Lesebequemlichkeit halber sei die Charakterisierung hier wiederholt:

Eigenschaften der Bereiche: situationsbasiert bei qualitativer Zeit; vollständige Bereichsinformation bei Planung und Ausführung; kontextfreie, deterministische Aktionen; konsistente, operationalisierte Ziele; Anwendungsbereich bei Planung und Ausführung statisch; ein einziger Planer; kein Fallwissen.

Anforderungen an Planer und Pläne: korrekt bezüglich des Merkmals, korrekte und vollständige Pläne zu erzeugen; oft Optimalität von Plänen bzgl. Zahl oder Kostensumme der Operatoren; Ressourcenabhängigkeit und Fallbasierung ausgeschlossen; Planer-Vollständigkeit nicht gefordert.

Der historisch wohl einflußreichste Planer ist STRIPS [Fikes und Nilsson, 1971]; weitere wichtige Planer mit entsprechenden Arbeiten dazu sind NONLIN [Tate, 1977], NOAH [Sacerdoti, 1977], TWEAK [Chapman, 1987] und neuerdings die Prozedur SNLP [McAllester und Rosenblitt, 1991]. Das Buch [Hertzberg, 1989] gibt einen ausführlichen Überblick über Techniken und Probleme im klassischen Planen.

Die Grundidee beim klassischen Planen ist, daß ein Plan eine geordnete Menge von Handlungen angeben soll, die der Reihe nach ausgeführt werden sollen. Bezeichnet man die formale Entsprechung einer Handlung als Operatorinstanz,[9] ist ein klassischer Plan formal also folgendermaßen definiert:

Definition 1 (Plan, „klassisch") *Ein Plan ist ein Paar* $\langle O, \prec \rangle$ *bestehend aus einer Menge von Operatorinstanzen* O *und einer strikten (möglicherweise partiellen) Ordnung* $\prec$ *auf* O.

Ist die Ordnung $\prec$ lediglich partiell, ist das so zu interpretieren, daß der Plan als Kurzform für die Menge der Pläne mit allen entsprechenden Linearisierungen der Operatoren steht. Ein solcher Plan mit einer nicht linear geordneten Operatormenge heißt *nichtlinearer Plan;* vollständig geordnete Pläne heißen *linear.* Enthält die Ordnung $\prec$ für irgendein $o \in O$ die Information, daß $o \prec o$, soll also o angeblich sein eigener Vorgänger sein, so heißt $\prec$ *inkonsistent.*

Der Definition kann man bereits ansehen, wie sie Einschränkungen des Zeitmodells ausnutzt, die das klassische Planen charakterisiert: beispielsweise degeneriert Zeit zum Vorher und Nachher von Handlungen. Weiterhin ist offensichtlich, daß diese Definition eine Einschränkung der Begriffsbestimmung 1 ist, und zwar eine sehr rigide.

Definition 1 legt die syntaktische Struktur von Plänen fest, erlaubt aber offenbar noch nicht, zwischen „sinnvollen" und „sinnlosen", „korrekten" und „inkorrekten" Plänen zu unterscheiden. Dazu müssen die Operatoren und die Pläne mehr Struktur bekommen.

Wie bereits in Abschnitt 2 gebe ich für Operatoren Vor- und Nachbedingungen an, also Bedingungen, die gelten müssen, damit der Operator ausgeführt werden kann bzw. Bedingungen, die gelten, nachdem er ausgeführt wurde. Da vorausgesetzt ist, daß Operatoren keine kontextabhängigen Wirkungen haben, können die Nachbedingungen unbedingt sein.

Um die Probleme beim Erschließen von Wandel auszuschließen, mache ich die folgende pragmatische Zusatzannahme: Operatoren haben *genau* die in ihren Nachbedingungen angegebenen Wirkungen. Diese Annahme zieht nach sich, daß keine Unterscheidung zwischen „direkten" Effekten und Schneeballeffekten gemacht werden kann bzw. alle Schneeballeffekte als direkte Effekte angegeben werden müssen. Da sie historisch zuerst im Planer STRIPS gemacht wurde, wird sie oft als STRIPS-Annahme bezeichnet.

Legt man zusätzlich fest, daß Situationen im Anwendungsbereich durch Mengen (d.h. Konjunktionen) von Grundatomen dargestellt sind und Operatorinstanzen das Gültigwerden bzw. Ungültigwerden von Grundatomen bewirken, dann erlaubt die STRIPS-Annahme die folgende rechnerisch einfache Bestimmung der Nachfolgesituation einer Operatoranwendung: die ungültig gewordenen Merkmale werden aus der Situationsbeschreibung herausgenommen und die neu gültig gewordenen hinzugefügt. Die folgende Definition macht dieses Verfahren präzise; dieses ist wiederum zuerst in STRIPS verwendet worden, weshalb ich diese Form von Operatoren als STRIPS-Operatoren bezeichne.

Definition 2 (STRIPS-Operator) *Eine* STRIPS-*Operator-Instanz ist ein Tripel* $\langle V, W, D \rangle$, *wobei* V, W, D *Mengen von Grundatomen sind, entsprechend der Vorbedingung, den wegfallenden Fakten und den dazukommenden Fakten.*

[9] Im folgenden werden Handlungen als solche (*action types*) über Operator*schemata* definiert, Einzelhandlungen (*action token*) über Operator*instanzen;* ein Plan kann also mehrere gleichartige Operatorinstanzen desselben Operatorschemas enthalten. In der Regel spreche ich kurz von *Operatoren,* wenn im Kontext klar ist, ob jeweils ein Schema oder eine Instanz gemeint ist.

$$\begin{array}{rrl}
\textbf{NIMM}(x): & \textbf{Vorb.:} & \{\mathsf{Tisch}(x), \mathsf{Frei}(x), \mathsf{Hand}(\textsc{nil})\} \\
& \textbf{Weg:} & \{\mathsf{Tisch}(x), \mathsf{Frei}(x), \mathsf{Hand}(\textsc{nil})\} \\
& \textbf{Dazu:} & \{\mathsf{Hand}(x)\}
\end{array}$$

$$\begin{array}{rrl}
\textbf{SETZE}(x): & \textbf{Vorb.:} & \{\mathsf{Hand}(x)\} \\
& \textbf{Weg:} & \{\mathsf{Hand}(x)\} \\
& \textbf{Dazu:} & \{\mathsf{Tisch}(x), \mathsf{Frei}(x), \mathsf{Hand}(\textsc{nil})\}
\end{array}$$

$$\begin{array}{rrl}
\textbf{STAPLE}(x,y): & \textbf{Vorb.:} & \{\mathsf{Hand}(x), \mathsf{Frei}(y)\} \\
& \textbf{Weg:} & \{\mathsf{Hand}(x), \mathsf{Frei}(y)\} \\
& \textbf{Dazu:} & \{\mathsf{Hand}(\textsc{nil}), \mathsf{Frei}(x), \mathsf{Auf}(x,y)\}
\end{array}$$

$$\begin{array}{rrl}
\textbf{TRENNE}(x,y): & \textbf{Vorb.:} & \{\mathsf{Hand}(\textsc{nil}), \mathsf{Frei}(x), \mathsf{Auf}(x,y)\} \\
& \textbf{Weg:} & \{\mathsf{Hand}(\textsc{nil}), \mathsf{Frei}(x), \mathsf{Auf}(x,y)\} \\
& \textbf{Dazu:} & \{\mathsf{Hand}(x), \mathsf{Frei}(y)\}
\end{array}$$

Abbildung 1: Schemata der Operatoren für die Blockwelt. Die intuitive Bedeutung der Operatoren: **NIMM**(x): Nimm x vom Tisch auf; **SETZE**(x): Setze x auf den Tisch ab; **STAPLE**(x, y): Staple x aus der Hand auf y, **TRENNE**(x, y): Nimm x von y ab und in die Hand.

Sei S die Beschreibung einer Situation, also eine Menge von Grundatomen. Das Ergebnis der Anwendung der STRIPS*-Operator-Instanz $o = \langle V, W, D\rangle$ in S ist definiert als:*

$$S' = \begin{cases} S & \textit{falls } V \not\subseteq S \\ (S \setminus W) \cup D & \textit{sonst} \end{cases}$$

Wie man sieht, bestehen Situationen hier lediglich aus der Beschreibung der definitiv wahren Fakten; man kann sie (und entsprechend die Operatoren) auch ergänzen um eine Menge von Grundatomen, die explizit *nicht* gelten. Das tut beispielsweise Bylander [1991], der Komplexitätsergebnisse für das resultierende „aussagelogische Planen“ beschreibt. Als Beispiele für STRIPS-Operatoren sind in Bild 1 Schemata für Operatoren aus der Blockwelt angegeben, wie wir sie auch bereits in Abschnitt 2 betrachtet haben. Operator-*Schema* soll besagen: Wenn man die Schema-Variablen x, y jeweils vollständig durch Terme, also beispielsweise die Namen von Blöcken wie A, B, C ersetzt, dann erhält man die Operator-Instanzen, die man in Plänen verwenden kann.[10]

Pläne werden erzeugt, um Probleme zu lösen. In der vereinfachten Sichtweise hier besteht ein Problem aus drei Komponenten: der Zustand, in dem man ist, den Merkmalen, die man gerne hätte, und den Handlungen, die man grundsätzlich zur Verfügung hat, um den aktuellen Zustand zu verändern. Das entsprechende syntaktische Konstrukt zur

[10] Zwei zusätzliche Bemerkungen am Rande. (1) Die angegebenen Operatorschemata sind unvollständig in dem Sinne, daß sie nicht verbieten, beispielsweise einen Block auf sich selber zu stapeln, also **STAPLE**(A, A) anwenden zu wollen – das muß zusätzlich ausgeschlossen werden. (2) Instanzen der Operatorschemata können natürlich ihrerseits Variablen enthalten, nämlich *Objekt*-Variablen, die von den Schema-Variablen in den Definitionen zu unterscheiden sind.

Beschreibung von Problemen ist entsprechend definiert:[11]

Definition 3 (STRIPS-Problembeschreibung) *Eine STRIPS-Problembeschreibung ist ein Tripel $\langle S, O, F \rangle$; dabei sind*

S *eine Menge von Grundatomen (Startsituation),*

O *eine Menge von STRIPS-Operatorschemata und*

F *eine Menge von Grundatomen (Zielmerkmale)*

Statt „Problembeschreibung" verwende ich oft kürzer „Problem", wenn klar ist, daß nicht das intuitive Problem in der Anwendungswelt gemeint ist, sondern eben seine Repräsentation für den Planer.

Ein Plan, der die Lösung eines Problems beschreibt, muß nun mindestens die folgenden Bedingungen erfüllen:

- er muß im Startzustand anwendbar sein;
- am Ende seiner Ausführung müssen die Zielmerkmale gelten;
- alle seine Operatoren müssen anwendbar sein.

Die folgenden Definitionen drücken das aus bzw. stellen das Vokabular zur Verfügung, es auszudrücken.

Definition 4 (Erzeuger, Nutzer, Zerstörer) *Sei $o = \langle V, W, D \rangle$ ein Operator und m ein Merkmal.*
o ist ein Erzeuger von m, gdw. $m \in D$.
o ist ein Nutzer von m, gdw. $m \in V$.
o ist ein Zerstörer von m , gdw. $m \in W$.

Die folgende Definition der Wohlgeformtheit eines Plans ist geleitet durch die folgende Intuition: alle Merkmale, die Operatoren im Plan als Vorbedingung brauchen, müssen vorher erzeugt worden sein – und sei es durch die Startsituation. Dabei setze ich voraus, daß die Startsituation in einem Plan syntaktisch durch einen Pseudo-Operator $\mathcal{I}$ dargestellt wird („Pseudo", weil er nicht tatsächlich ausgeführt wird), der als eindeutig erster Operator gemäß $\prec$ eingeordnet ist, der keine Vorbedingungen hat und der als Nachbedingung die Menge der Merkmale der Startsituation hat. Desgleichen stelle ich die Zielmerkmale als Pseudo-Operator $\mathcal{G}$ am Plan-Ende dar, der diese Zielmerkmale als Vorbedingungen hat und ohne Nachbedingungen ist.

Definition 5 (Wohlgeformtheit) *Der (klassische) Plan $\Pi = \langle O, \prec \rangle$ ist wohlgeformt, gdw. für alle Merkmale m und jeden Operator $n \in O$, der m nutzt, es einem m-Erzeuger $e \in O$ gibt, so daß gilt: $e \prec n$.*

Wohlgeformtheit ist notwendig dafür, daß ein Plan „fertig" ist, aber nicht hinreichend, denn ein einmal erzeugtes Merkmal kann ja möglicherweise anschließend wieder zerstört worden sein. Um auszudrücken, welcher Operator welches Merkmal für welchen anderen Operator erzeugt, benutze ich das Konzept der Abhängigkeit von Operatoren bzgl. eines Merkmals:

[11] Die formale Beschreibung des historisch tatsächlichen STRIPS erfordert eine weitere Komponente, die *wesentlichen* Fakten [Lifschitz, 1987]. Davon abstrahiere ich hier.

Definition 6 (Abhängigkeit) *Sei* $\Pi = \langle O, \prec \rangle$ *ein Plan, seien* $e, n \in O$*, und sei* m *ein Merkmal.*

$\delta_\Pi = \langle e, m, n \rangle$ *ist eine Abhängigkeit zwischen* e *und* n *bezüglich* m *in* Π*, gdw.*

- e *ist Erzeuger von* m*;*
- n *ist Nutzer von* m*;*
- $e \prec n$*; und*
- *kein* $o \in O$ *ist Erzeuger von* m*, für den gilt* $e \prec o \prec n$*.*

Damit ein wohlgeformter Plan „fertig“ ist, müssen also alle Operator-Vorbedingungen durch Abhängigkeiten „erreicht“ werden. Doch auch das ist immer noch nicht hinreichend, denn es kann sein, daß eine Abhängigkeit $\langle e, m, n \rangle$ einen m-Zerstörer umschließt bzw. nach einer entsprechenden Vervollständigung oder Erweiterung der Planordnung einen solchen m-Zerstörer umschließen könnte. Diese Konstellation heißt Konflikt, und sie ist in drei Schritten so definiert:

Definition 7 (Erweiterung) *Sei* $\prec$ *eine strikte partielle Ordnung auf einer Menge* M*.* $\prec' \supseteq \prec$ *ist eine Erweiterung von* $\prec$*, gdw.* $\prec'$ *ebenfalls eine strikte partielle Ordnung auf* M *ist.*

Definition 8 (umschließen, Einfügung) *Sei* $\Pi = \langle O, \prec \rangle$ *ein Plan. Die Abhängigkeit* $\delta_\Pi = \langle e, m, n \rangle$ *umschließt einen Operator* $o \in O$*, gdw. es existiert eine Erweiterung* $\prec'$ *von* $\prec$*, so daß* $e \prec' o \prec' n$*.*

$\prec'$ *heißt dann Einfügung von* o *in* δ_Π*.*

Definition 9 (Konflikt) *Ein Plan* $\Pi = \langle O, \prec \rangle$ *enthält einen Konflikt zwischen einem Operator* $z \in O$ *und einer Abhängigkeit* $\delta_\Pi = \langle e, m, n \rangle$*, gdw.*

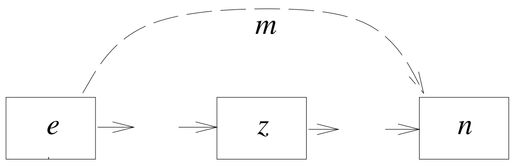
- z *zerstört* m*, und*
- *es existiert eine Einfügung* $\prec'$ *von* z *in* δ_Π*, so daß für alle* $o \in O$ *gilt: wenn* $z \prec' o \prec' n$*, dann erzeugt* o *nicht* m*.*

Zur Veranschaulichung sind die möglichen Typen von Konflikten in nichtlinearen Plänen in Bild 2 graphisch dargestellt.

Das technische Kriterium dafür, daß ein Plan „fertig“ ist, ist nun, daß er wohlgeformt und konfliktfrei ist. Die Grundidee einer Basisprozedur zum Erstellen klassischer, nichtlinearer Pläne besteht nun darin, mit dem Plan-„Embryo“ bestehend aus den Pseudo-Operatoren $\mathcal{I}$ und $\mathcal{G}$ anzufangen und solange Operatoren einzufügen bzw. die Operatorordnung zu verschärfen, bis der resultierende Plan wohlgeformt und konfliktfrei ist; man überzeugt sich leicht davon, daß in diesem Fall alle Operatoren in jeder vollständigen Linearisierung des Plans anwendbar sind und daß am Ende die Zielmerkmale gelten. Eine solche Basisprozedur ist in Bild 3 mit der Prozedur $\text{NLP}_{\text{GRUND}}$ angegeben.

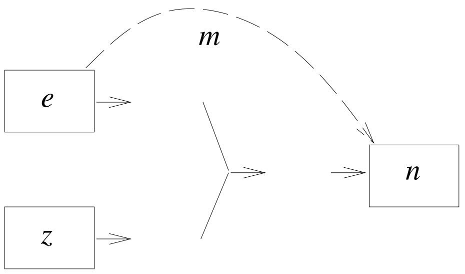
Das tiefgestellte GRUND im Namen $\text{NLP}_{\text{GRUND}}$ weist darauf hin, daß in den Plänen, die die Prozedur erstellen kann, an allen Stellen voll instanzierte Grundmerkmale vorkommen. Es wäre grundsätzlich denkbar, in den Fällen Objektvariablen als Operatorargumente zu benutzen, wo ein konkreter Block nicht festgelegt zu werden braucht; um beispielsweise das Merkmal Hand(NIL) zu erzeugen, wenn gerade A gehalten wird, könnte man den Operator

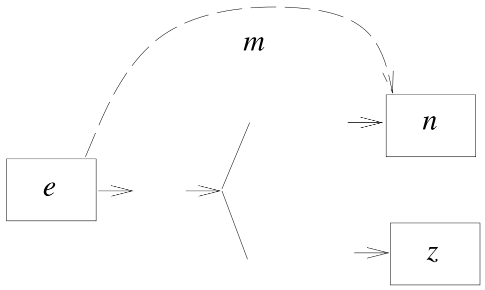
Linearkonflikt: $e \prec z$ und $z \prec n$



Gabelkonflikte:

$z \prec n$, $e \not\prec z$ und $z \not\prec e$ (linke Gabel) oder $e \prec z$, $n \not\prec z$ und $z \not\prec n$ (rechte Gabel)





Parallelkonflikt: weder $z \prec n$ noch $e \prec z$

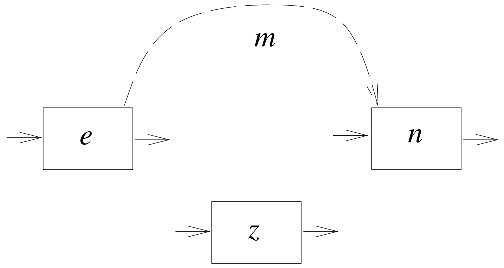


Abbildung 2: Die möglichen Topologien von Konflikten in nichtlinearen Plänen.

Prozedur $\text{NLP}_{\text{GRUND}}(\Pi)$

Eingabe: $\Pi = \langle O, \prec \rangle$: klassischer nichtlinearer Plan

Ausgabe: klassischer nichtlinearer Plan

---

0. **if** $\prec$ ist inkonsistent **then return fail**;
1. **if** Π wohlgeformt und konfliktfrei **then return** Π;
2. Wähle Operator $o \in O$ aus, der eine ungelöste Vorbedingung v hat;
3. **if** v ungelöst durch Konflikt zwischen z und $\langle e, m, o \rangle$
   then wähle eine der Alternativen

   Linearisieren I: $\Pi := (O, \prec \cup \{(z, e)\})$

   Linearisieren II: $\Pi := (O, \prec \cup \{(o, z)\})$

   Einordnen: Wähle einen zu z ungeordneten v-Erzeuger $e \neq e' \in O$;
   $\Pi := (O, \prec \cup \{(z, e'), (e', o)\})$

   Einfügen: Wähle einen v-Erzeuger $e' \notin O$;
   $\Pi := (O \cup \{e'\}, \prec \cup \{(z, e'), (e', o)\})$

   else wähle v-Erzeuger e' gemäß einer der Alternativen

   Einordnen: Wähle $e' \in O$, $e' \not\prec o$;
   $\Pi := (O, \prec \cup \{(e', o)\})$

   Einfügen: Wähle v-Erzeuger $e' \notin O$;
   $\Pi := (O \cup \{e'\}, \prec \cup \{(e', o), (\mathcal{I}, e')\})$

4. **return** $\text{NLP}_{\text{GRUND}}(\Pi)$

Abbildung 3: Eine Basisprozedur zum nichtlinearen klassischen Planen; Erklärung im Text.

$$\left\{\begin{array}{c}\mathsf{Tisch}(\text{A}), \mathsf{Tisch}(\text{B}),\\ \mathsf{Frei}(\text{B}), \mathsf{Frei}(\text{C}),\\ \mathsf{Auf}(\text{C},\text{A}), \mathsf{Hand}(\text{NIL})\end{array}\right\} \qquad \left\{\ \mathsf{Auf}(\text{A},\text{B}), \mathsf{Auf}(\text{B},\text{C})\ \right\}$$

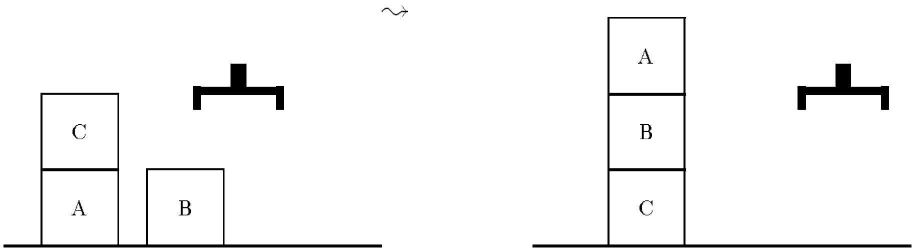


Abbildung 4: Start- und Zielmerkmale des Blockwelt-Problems „Sussman-Anomalie“ und ihre bildliche Darstellung. Der Name Sussman-Anomalie kommt daher, daß dieses Problem Sussmans [1975] Planer HACKER zu einem in gewissem Sinn anomalen Verhalten geführt hat.

STAPLE(A, x) einsetzen für ein später noch festzulegendes $x \neq \text{A}$. Es ist heute durchaus üblich, so vorzugehen, aber ich will hier nicht darauf eingehen. Planer, für die die Verwendung von Objektvariablen beschrieben ist, sind beispielsweise TWEAK [Chapman, 1987] und SIPE [Wilkins, 1988].

Der initiale Aufruf der Prozedur ist von der Form

$$\text{NLP}_{\text{GRUND}}(\langle\{\mathcal{I}, \mathcal{G}\}, \{(\mathcal{I}, \mathcal{G})\}\rangle),$$

wobei, wie gesagt, $\mathcal{I}$ und $\mathcal{G}$ die Pseudo-Operatoren sind, die der Startsituation und den Zielmerkmalen entsprechen. Durch die Vorschriften zum Einfügen von Operatoren in den enstehenden Plan ist gewährleistet, daß $\mathcal{I}$ und $\mathcal{G}$ immer die $\prec$-minimalen bzw. -maximalen Elemente bleiben. $\text{NLP}_{\text{GRUND}}$ beschreibt dann im Kern eine Suche durch den Suchraum der unvollständigen Pläne, wobei dieser Raum implizit gegeben ist durch den Initialplan als Wurzel und die möglichen Alternativen in Schritt 3 für alle in Schritt 2 auswählbaren Operatoren o.

In Schritt 0 wird geprüft, ob die Planordnung inkonsistent ist, also ob beispielsweise für ein $o \in O$ über den transitiven Abschluß von $\prec$ gilt, daß $o \prec o$. Das kann daher kommen, daß in späteren Schritten der Prozedur Operatoren lokal geordnet werden, ohne daß geprüft wird, ob diese Ordnung möglich ist. Tritt dieser Fall auf, wird die Prozedur zurückgesetzt. In der angegebenen Formulierung ist chronologisches Rücksetzen impliziert; das kann man natürlich effektiver zu implementieren versuchen.

Als Beispiel für die Verwendung von $\text{NLP}_{\text{GRUND}}$ betrachten wir die STRIPS-Problembeschreibung, die besteht aus den Operatoren, deren Schemata in Abbildung 1 angegeben sind, und aus den Start- und Zielmerkmalen wie in Abbildung 4 dargestellt.

Abbildung 5 zeigt einen möglichen Zwischenzustand der Planung und den fertigen Plan, wie $\text{NLP}_{\text{GRUND}}$ sie erzeugen könnte. Der fertige Plan ist vollständig geordnet; das ist bezogen auf $\text{NLP}_{\text{GRUND}}$ ein Sonderfall, der darin begründet ist, daß für dieses Problem keine Ausführungsalternativen von Operatoren existieren.

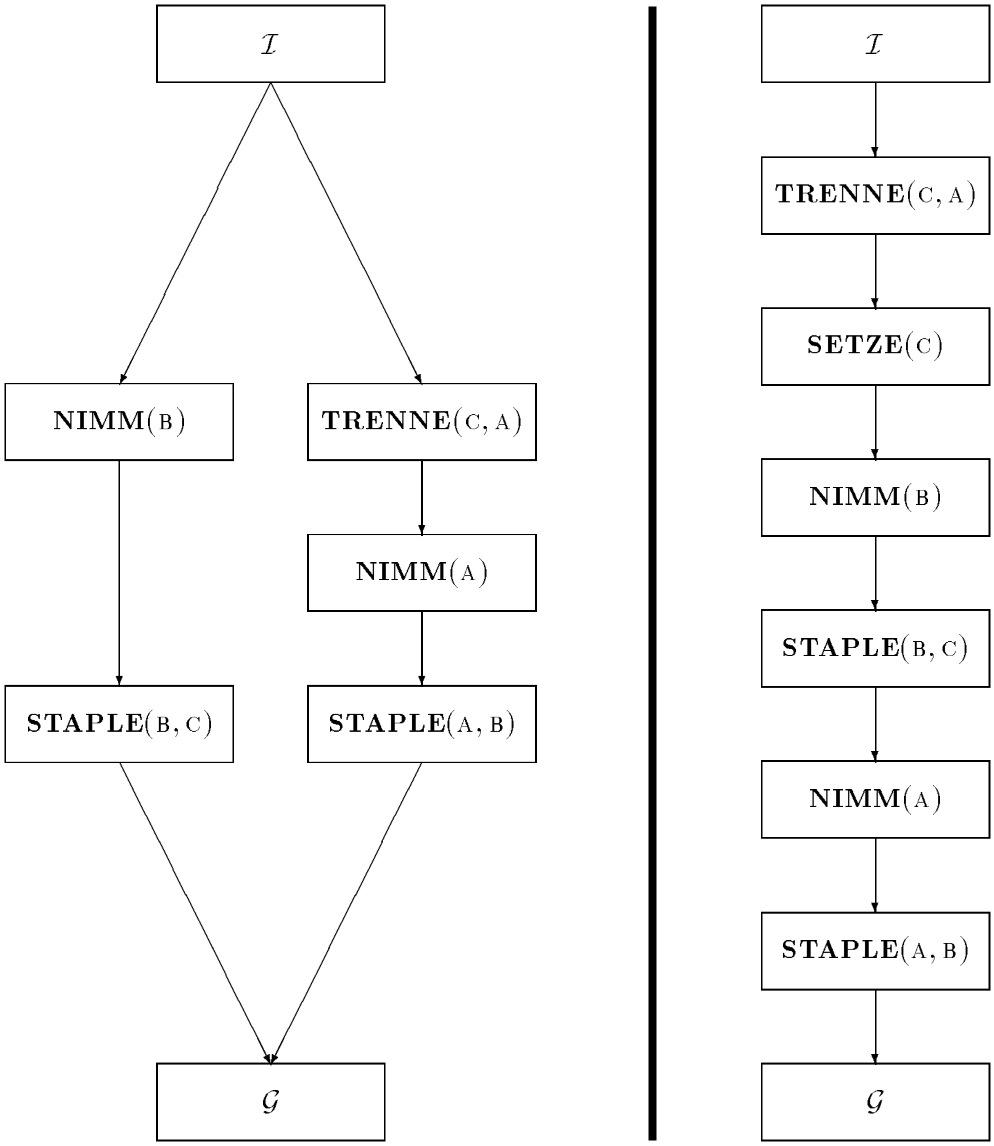


Abbildung 5: Möglicher Zwischenplan (links) und Endplan (rechts) von $\text{NLP}_{\text{GRUND}}$ für die Sussman-Anomalie. Operatoren sind dargestellt durch Kästchen, die Präzedenzordnung durch Pfeile. Der Zwischenplan ist wohlgeformt; aber er ist nicht konfliktfrei: Konflikte gibt es beispielsweise bezüglich der Merkmale Hand(NIL) und Frei(B).

Die Abhängigkeitsstruktur zwischen den Operatoren im Plan ergibt sich nach NLP$_{\text{GRUND}}$ in dem folgenden Sinn zufällig: Wurde ein Operator e eingesetzt, um das Merkmal m als Vorbedingung für einen anderen Operator n zu erzeugen, ergab sich also unmittelbar nach Einsetzen von e die Abhängigkeit $\langle e, m, n\rangle$, so ist nicht sicher, daß im endgültigen Plan es immer noch e ist, der m für n erzeugt. Es ist möglich, daß ein später eingesetzter m-Erzeuger e' zufällig oder bewußtermaßen (etwa wenn sich bezüglich $\langle e, m, n\rangle$ ein Konflikt ergeben hatte) der letztendliche m-Erzeuger für n wird. Die Abhängigkeiten innerhalb eines Planes wachsen also bei der Vervollständigung des Plans nicht monoton mit, sondern können wieder verschwinden. Das ist für die Prozedur insofern ohne Belang, als die Abhängigkeiten nicht explizit verwaltet werden, sondern sie ergeben sich zumindest konzeptuell in jedem Durchlauf wieder neu.[12]

Gerade neuere Arbeiten zu Grundlagen des klassischen nichtlinearen Planens in Folge der Prozedur SNLP [McAllester und Rosenblitt, 1991] untersuchen eine explizite Verwaltung der Abhängigkeiten und erlauben *nicht,* einmal festgelegte Abhängigkeiten wieder zu brechen: hat man sich einmal auf die Abhängigkeit $\langle e, m, n\rangle$ festgelegt, so muß sie auch im fertigen Plan enthalten sein. Dadurch wächst also die Menge der Abhängigkeiten in einem zu entwickelnden Plan monoton. Diese Strategie ist der Schlüssel zur *Systematizität* von SNLP, was in etwa bedeutet, daß ein Teilplan, der im Suchraum auf einem Pfad vom Initialplan zu einer Lösung liegt, nicht auf zweierlei Weise erzeugt werden kann. Für weitere Information sei auf [McAllester und Rosenblitt, 1991] verwiesen. Übrigens wächst in NLP$_{\text{GRUND}}$ die *Operatormenge* bei der Entwicklung eines Plans monoton, denn nirgends wird ein einmal in O aufgenommener Operator wieder entfernt. Das ist einfach eine Konvention, die sich im klassischen Planen als sinnvoll erwiesen hat. Theoretisch spräche nichts dagegen, beispielsweise Konflikte auch durch entfernen des zerstörenden Operators zu lösen.

Schritt 2 verwendet implizit die *Unabhängigkeitsannahme,* die besagt, daß mehrere an einer Stelle des Plans offene Operator-Vorbedingungen unabhängig voneinander erfüllt werden können. Diese Annahme ist heuristisch, kann also im Einzelfall falsch sein. Die Blockwelt eignet sich als didaktisch motivierter Demo-Anwendungsbereich fürs klassische Planen unter anderem deshalb so gut, weil die Unabhängigkeitsannahme hier häufig verletzt ist und sich Mechanismen zur Behandlung der resultierenden Plan-Konflikte ausführlich studieren lassen.[13]

In ihren Schritten 2 und 3 fällt auf, daß die Prozedur NLP$_{\text{GRUND}}$ mit (nichtdeterministischen) Auswahlen arbeitet. Das ist natürlich ein ausschließlich für die theoretische Betrachtung einsetzbarer Trick, um den Prozedurtext bündig aufschreibbar zu machen und dennoch alle möglichen Alternativen anzugeben, die an den entsprechenden Stellen grundsätzlich betrachtet werden müssen. Jeder Planer, der auf einem konventionellen Rechner entsprechend NLP$_{\text{GRUND}}$ ablaufen soll, muß diese nichtdeterministischen Auswahlpunkte in deterministische Auswahlen übersetzen – und tatsächlich liegt die Kunst, für einen gegebenen Anwendungsbereich einen Planer mit akzeptablen Laufzeiten zu schrei-

[12] Natürlich kann man das effizienter implementieren, indem man beim Übergang von einer Planversion auf die nächste die Abhängigkeiten im wesentlichen vererbt und nur Änderungen berücksichtigt. Solche Implementierungsaspekte seien aber hier ausgeblendet.

[13] In der Literatur, beispielsweise auch in [Hertzberg, 1989], wird die Unabhängigkeitsannahme oft Linearitätsannahme (engl. *linearity assumption* oder *linear assumption*) genannt. Um Verwirrung mit der Linearität bzw. Nichtlinearität der Operatorordnung zu vermeiden, verwende ich den anderen Namen.

ben, ganz wesentlich auch darin, diese Auswahlen jeweils „schlau“ zu treffen.

Was „schlau“ heißt, ist allgemein nicht zu sagen, sondern ist mindestens vom Anwendungsbereich abhängig. Es gibt jedoch eine Reihe von allgemeiner anwendbaren heuristischen Tricks, eine geschickte Auswahl zu unterstützen. Einige Beispiele:

- Unterteile Operator-Nachbedingungen in *Primär-* und *Sekundär*effekte, mit der Interpretation, daß zur Erzeugung des Merkmals m nur solche Operatoren verwendet werden, die m primär bewirken. Würde man beispielsweise in **STAPLE**(x, y) genau das Merkmal $\mathsf{Auf}(x, y)$ als Primäreffekt markieren, kämen Operatoren der Art **STAPLE**$(\cdot, \cdot)$ nicht mehr in Betracht, um das Merkmal $\mathsf{Hand}(\text{NIL})$ zu erzeugen.
- Unterteile Vorbedingungen in *Anwendbarkeitsbedingungen* und *Unterziele,* mit der Interpretation, daß ein Operator nur dann eingesetzt wird, wenn an der entsprechenden Stelle im Plan seine Anwendbarkeitsbedingungen bereits gelten. Beispiel: Anwendbarkeitsbedingung für **TRENNE**(x, y) ist $\mathsf{Auf}(x, y)$ – mit anderen Worten: plane nicht, x von y abzunehmen, wenn es nicht eh schon draufsteht.
- Wenn mehrere Operatoren ein offenes Merkmal erzeugen könnten, dann setze denjenigen ein, der möglichst wenige offene Vorbedingungen hat und möglichst wenige Konflikte erzeugt.

Beispiel-Faustregeln für die Operatorauswahl speziell im Bereich Blockwelt sind

- Wenn der Block x auf dem Tisch steht und das Ziel ist, ihn in der Hand zu halten, dann wähle **NIMM**(x).
- Wenn ein Block in einem Turm enthalten ist und noch bewegt werden muß, dann setzte keine anderen Blöcke auf die Turmspitze.

Diese Heuristiken werden im allgemeinen die Planung vereinfachen. Andererseits sind sie nicht vollständig in dem Sinn, daß ihre strikte Befolgung dazu führen kann. daß ein Problem unlösbar wird, das sonst eine Lösung gehabt hätte. Alternativ oder komplementär kann man natürlich die allgemeinen Suchstrategien wie Breitensuche, Tiefensuche oder A* [Nilsson, 1982] zur Spezialisierung von $\text{NLP}_{\text{GRUND}}$ verwenden. In der Regel lassen sich mit diesen allgemeinen Strategien allein jedoch bei weitem keine Planer erzielen, die einigermaßen effizient laufen.

Dieser pessimistischen Einschätzung entspricht die Tatsache, daß das klassische Planen, und selbst noch Einschränkungen dieser schon von sich aus so eingeschränkten Art von Planung auch theoretisch hoch komplex sind. Hierzu sei nur ein Hinweis gegeben auf die Arbeit von Bylander [1991], in die $\text{NLP}_{\text{GRUND}}$ direkt hineinpaßt. Bylander untersucht das *aussagelogische Planen* (*propositional planning*), das mit dem klassischen Planen wie hier beschrieben identisch ist bis auf die Tatsache, daß Situationen aus *zwei* Mengen von Grundatomen bestehen, nämlich denen, die positiv gelten, und denen, die definitiv nicht gelten, und daß Operatoren entsprechend zweigeteilte Vorbedingungen haben: die Merkmale, die gelten müssen, und die, die nicht gelten dürfen. Offensichtlich sind STRIPS-Operatoren dann genau solche, bei denen die zweite Menge leer ist. Bei der Berechnung der Nach-Situation nach Anwendung eines Operators werden die dazukommenden Fakten nicht nur der Menge der gültigen Fakten hinzugefügt, sondern auch aus der Menge der falschen Fakten gelöscht; für die wegfallenden Fakten gilt Entsprechendes genau andersherum. Das Entscheidungsproblem PLANSAT besteht dann aus der Frage, ob es zu einer

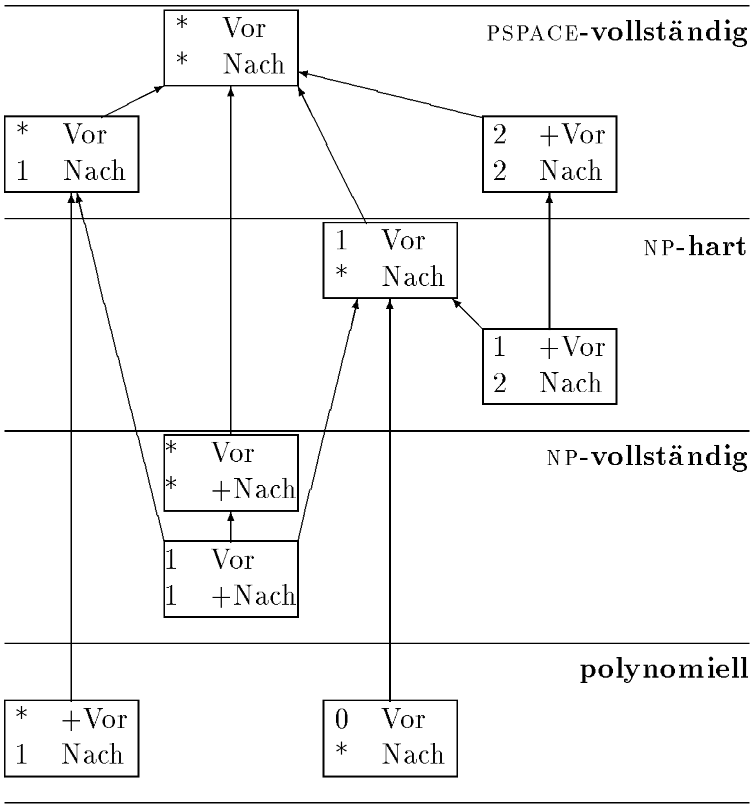


Abbildung 6: Komplexitätsergebnisse für propositionales Planen (nach [Bylander, 1991], leicht vereinfacht). „+Vor" bedeutet „positive Vorbedingung(en)"; „+Nach" entsprechend. „*" bedeutet „beliebig viele". Die Problemklasse unten links ist also zu lesen: Beliebig viele positive (und keine negativen) Vorbedingungen, 1 (positive oder negative) Nachbedingung; diese Problemklasse ist polynomiell lösbar. Die Pfeile zeigen von spezielleren in eine nächst allgemeinere Problemklasse.

gegebenen Problembeschreibung einen wohlgeformten und konfliktfreien Plan gibt, der eine Lösung des entsprechenden Problems beschreibt; die Größe eines solchen Problems entspricht der Summe der Operatorinstanzen und Merkmale.

Als erstes stellen wir fest, daß PLANSAT entscheidbar ist, sofern es nur eine endliche Menge von Objekten und Prädikaten im Anwendungsbereich gibt, also beispielsweise nur endlich viele Blöcke in der Blockwelt. Das liegt daran, daß der zugrundeliegende Zustandsraum dann endlich ist und man ihn im Prinzip vollständig untersuchen kann. Doch wie hoch ist die Berechnungskomplexität im schlimmsten Fall? Bylander hat hierzu unterschiedliche Einschränkungen des aussagelogischen Planens untersucht, die im wesentlichen dadurch entstehen, daß die Form der Operatoren eingeschränkt wird. Seine Ergebnisse sind in Bild 6 zusammengefaßt. Wie man sieht, sind bereits rigideste, praktisch gesehen völlig hoffnungslos eingeschränkte Versionen des aussagelogischen Planens bereits nicht mehr effizient berechenbar; zum Beispiel wird PLANSAT bereits NP-vollständig, wenn man für Operatoren nur 1 Vorbedingung und 1 positive Nachbedingung erlaubt. Für weitere Details sei auf die angegebene Arbeit von Bylander verwiesen. Klassisches Planen mit STRIPS-Operatoren, wie es $\text{NLP}_{\text{GRUND}}$ leistet, ist offensichtlich PSPACE-vollständig, da es die Klasse „2 +Vor, 2 Nach“ umfaßt.

Klassisches Planen ist also theoretisch hochkomplex. Wenn man, wie in TWEAK [Chapman, 1987], auch noch Objektvariablen erlaubt (was aus Anwendungsgesichtspunkten ausgesprochen sinnvoll scheint), dann wird die Frage der Lösbarkeit eines Planungsproblems gar unentscheidbar, wie Chapman gezeigt hat. Es kann also nicht *den* Satz von Heuristiken oder Strategien geben, der Planen in allen Fällen effizient macht.[14] Es gibt aber zwei Mittel der Strukturierung des Problembereichs bzw. eines Plans, die praktisch nützlich sein können und immer wieder verwendet wurden:

Vorbedingungselimination. Die Idee ist, die Merkmale des Anwendungsbereichs in Stufen einzuteilen. Pläne werden dann erzeugt, indem zunächst nur die wichtigsten Vorbedingungen berücksichtigt werden, dafür ein wohlgeformter konfliktfreier Plan erzeugt wird, dieser dann zur Berücksichtigung der nächst wichtigen Merkmale ausgebaut wird und so fort, bis alle Merkmale berücksichtigt sind. Wenn die Wichtung den Planungsaufwand geschickt auf die unterschiedlichen Stufen verteilt und wenn Pläne höherer Stufen in niedrigere ausbaubar sind, können sich erhebliche Effizienzgewinne ergeben. Der Schlüssel zum Erfolg liegt natürlich in der Wahl einer geschickten Merkmalswichtung, wofür es kein Patentrezept gibt.

Vorbedingungselimination wurde erstmals in ABSTRIPS [Sacerdoti, 1974] verwendet[15]; [Tenenberg, 1991] kann als Einstieg in neuere Arbeiten zum Thema dienen.

Virtuelle Operatoren. Die Idee ist, Operatoren zu definieren und in Plänen zu verwenden, die nicht unmittelbar ausführbar sind, sondern hinter denen sich Teilpläne verbergen, die ihrerseits möglicherweise noch ausgebaut werden müssen. Ein Plan kann damit also oberflächlich gesehen „fertig“ sein, aber noch einen virtuellen Operator enthalten, der in einem nächsten Planungsschritt durch den entsprechenden Teilplan ersetzt wird, wodurch der Gesamtplan wieder lückenhaft wird. Der praktische Effizienzgewinn fürs Planen ergibt sich dadurch, daß virtuelle Operatoren Standard-Teilpläne für Standard-Unterprobleme formulieren können, die der Planer nicht jedesmal von Grund auf neu finden muß. Zum Beispiel könnte man in der Blockwelt einen virtuellen Operator **BAUE-DREIER**(x, y, z) zum Bau eines Dreierturms aus x, y, z formulieren, dessen Expansion angibt, daß die beiden elementaren Operatoren **STAPLE**(y, z) und **STAPLE**(x, y) in dieser Reihenfolge ausführen, den Turm also von unten her aufbauen muß – was andernfalls jedesmal von Neuem durch Konflikterkennung und -auflösung zwischen den beiden Elementaroperatoren entdeckt werden müßte.

[14] Es sei denn, dieser Satz von Heuristiken ist geeignet zu zeigen, daß P = NP.

[15] Dort und anderswo wird Vorbedingungselimination als *Situationsabstraktion* bezeichnet. Diese Bezeichnung ist insofern irreführend, als die technische Stufung von Merkmalen mit der intuitiven Vorstellung von Abstraktion wenig zu tun zu haben scheint, auch wenn diese bei der Entwicklung der Vorbedingungselimination ursprünglich Pate gestanden haben mag.

Virtuelle Operatoren wurden erstmals in NOAH [Sacerdoti, 1977] verwendet[16]; ein neuerer Planer mit virtuellen Operatoren ist SIPE [Wilkins, 1988].

Für ausführlichere Beschreibungen verweise ich auf die angegebene Originalliteratur oder [Hertzberg, 1989].

Bei der Charakterisierung des klassischen Planens hatten wir bereits gesehen, daß nicht gefordert ist, daß ein Planer *alle* Pläne liefert, die ein Problem lösen. Das ist natürlich ebenfalls eine Maßnahme, die praktische Komplexität des klassischen Planens zu verringern. Eine extremere Form der Unvollständigkeit wäre, Laufzeitschranken vorzugeben, innerhalb derer entweder ein Plan gefunden wird, oder das Problem wird als unlösbar angesehen. Das weist bereits in Richtung von *anytime*-Prozeduren, die ich im nächsten Abschnitt ansprechen werde. $\text{NLP}_{\text{GRUND}}$ ist darauf angelegt, *einen* Plan zu finden, kann aber offenbar entsprechend modifiziert werden, auch mehrere Pläne zu erzeugen oder Laufzeitschranken zu beachten.

Gleichsam dual zum Aufgeben der Planer-Vollständigkeit ist es, Korrektheit aufzugeben, also möglicherweise auch Pläne zurückzugeben, die nicht wohlgeformt oder nicht konfliktfrei sind. Das ist mit der Philosophie des klassischen Planens jedoch gänzlich unverträglich, obwohl es nicht völlig unsinnig erscheint. Ich werde darauf in Abschnitt 6 zurückkommen.

4 Nicht-klassische Planung

Das klassische Planen ist charakterisiert durch eine ganz bestimmte Auswahl der Merkmale von Anwendungsbereichen und Anforderungen an Pläne und Planer, die ich in Abschnitt 2.2 aufgezählt habe; als Faustregel gilt: das klassische Planen vereinfacht alles maximal. Das ist sinnvoll, um mit dem Studium der Problemklasse Planen zu beginnen, und naheliegend, wenn Anwendungsbereiche in Einzelfällen tatsächlich die entsprechende Struktur haben oder wenn man es selber in der Hand hat, sie in diese Struktur zu bringen. Im allgemeinen Fall ist die Welt aber nicht so einfach, und man plant unter anderen Randbedingungen.

Dazu gibt es in der Literatur inzwischen eine Fülle von Arbeiten. Im Rahmen dieses einführenden Überblicks können daraus nur Einzelthemen punktuell berücksichtigt werden. Daher will ich hier für Themen, die nicht weiter behandelt werden, wenigstens Hinweise auf weiterführende Literatur geben:

Behandlung von Zeit hat eine relativ lange Tradition im KI-Planen. Bearbeitet werden wesentlich die folgenden Themen:

- Verarbeitung quantitativer Zeitinformation, um Pläne für die Ausführung beispielsweise auf Uhrzeiten abzubilden; ein klassischer Text dazu ist [Vere, 1983].
- Modellierung des Effekts von Nebenläufigkeit von Handlungen („Wie unterscheidet sich der Effekt nebenläufiger Handlungen von der Summe der Einzeleffekte?“); eine Einführung darein gibt [Pelavin, 1991].

[16] Dort und anderswo werden virtuelle Operatoren als *abstrakte Operatoren* bezeichnet. Diese Bezeichnung ist insofern irreführend, als die Technik des Einsetzens und Expandierens von Teilplänen mit der intuitiven Betrachtung von Handlungen in unterschiedlichen Abstraktionsgraden wenig zu tun zu haben scheint, auch wenn diese bei der Entwicklung der Vorbedingungselimination ursprünglich Pate gestanden haben mag.

Sinnvollerweise wird das Schließen über Zeit in einem Planer eingebettet in allgemeinere Mechanismen zum Schließen über Zeit. Beispiel für ein allgemeineres Verfahren zur Verarbeitung quantitativer Zeitinformation ist das System TMM [Dean und McDermott, 1987]; ein Beispiel für Planen mit einem reichen Modell von Zeitinformation (nämlich den Allen-Relationen) gibt [Allen, 1991]. Probleme der Nebenläufigkeit werden insbesondere dann wichtig, wenn verteilte Ausführung von Plänen durch mehrere Akteure betrachtet wird.

Verteiltes Planen ist ein Teilbereich des Gebiets „Verteilte KI“ (VKI). Grundidee ist, die Arbeit, einen Plan zu erstellen, auf mehrere (Teil-)Planer aufzuteilen, die jeweils Einzelteile des Gesamtplans erstellen, die schließlich zusammengesetzt werden. Motive für Verteilung sind beispielsweise Robustheit des Gesamtsystems, Spezialisierung einzelner Planer auf bestimmte Arten von Teilproblemen, dadurch möglicherweise entstehende Kostenreduzierung, und nicht zuletzt die Tatsache, daß in der Welt Informationen und Fähigkeiten faktisch verteilt vorliegen. Gegenüber dem Planen mit einem einzelnen Planer entstehen dadurch aber natürlich neue Anforderungen (zusätzlich zu denen im unverteilten Planen), zum Beispiel Zerlegung des Gesamtproblems, Verteilung der Teilprobleme auf die Planer, Kommunikation zwischen den Planern, Zusammensetzen der Teilpläne zum Gesamtplan.

[Bond und Gasser, 1988] ist eine Sammlung von historisch wichtigen Texten und [von Martial, 1993] ein neuerer Überblick über VKI im allgemeinen und über verteiltes Planen im besonderen; [Hertzberg, 1989, Abschnitt 6.4] gibt ein Beispiel für verteiltes Planen (von verteilt ausführbaren Plänen) aufsetzend auf Techniken des klassischen Planens.

Wiederverwenden von Plänen oder Planteilen zur Lösung von gleichen oder ähnlichen Problemen, Pläne also gewissermaßen zu „lernen“, ist eine naheliegende Idee, um die Effektivität des Planens im selben Anwendungsbereich zu steigern. Der erste Ansatz dazu wurde bereits im Kontext von STRIPS gemacht [Fikes *et al.*, 1972], wo Pläne zu „Makro-Operatoren“ verallgemeinert wurden, die zur weiteren Planung wie die normalen Bereichsoperatoren in Pläne eingesetzt werden konnten; eine verbesserte Variante dieses Vorgehens schildert [Minton, 1985]. Eine konzeptionell weitergehende Art von Wiederverwendung ist das *fallbasierte Planen.* Die Idee dabei ist, daß Planen grundsätzlich darin besteht, zur Lösung von Problemen alte Pläne wiederzuverwenden bzw. anzupassen, die für „ähnliche“ Probleme früher erstellt wurden, und nur in Ausnahmefällen tatsächlich neu zu planen. Das erfordert folglich andere Planungsprozeduren, außerdem die Verwaltung einer Bibliothek alter Pläne und die Definition eines geeigneten Ähnlichkeitsmaßes auf Plänen. Bekannte Planungssysteme in dieser Denkrichtung sind PLEXUS [Alterman, 1986] und CHEF [Hammond, 1989], wobei das letztere System den Anspruch hat, tatsächlich auch die Bereichsrepräsentation durch Lernen in begrenztem Maß zu erweitern.

Handeln „aus der Situation“ (*situated action*) bezeichnet eine Sichtweise der Steuerung von Handlungen, die dem Planen geradezu entgegengesetzt ist.[17] Ausgangsvoraussetzung ist die Annahme, daß Akteure nicht dadurch zielgeleitet handeln, daß

[17] Bei einigen Autoren aus dem „Lager der Situierten“ nimmt die Betonung dieses Gegensatzes fast Weltanschauungs- oder Ideologie-Dimensionen an.

sie in symbolischer Form vorliegende Pläne abarbeiten, sondern daß sie aus Erfahrung Regeln entwickelt haben, die primär ihr Handeln abhängig von der aktuell vorliegenden Situation steuern. Eventuell vorliegende symbolische Pläne haben eher sekundäre Bedeutung als *eine* Komponente der aktuellen Situation, die das Handeln *mit*bestimmt, ohne es (wie beispielsweise im klassischen Planen) vollständig zu determinieren. Ein Schlüsselaufsatz für diese Sichtweise ist [Agre und Chapman, 1987]; [Ringle, 1993] enthält eine ausführliche Diskussion von Sinn und Unsinn, Für und Wider der Sichtweise Situationsbasierung. [Ginsberg, 1989] enthält komplexitätstheoretische Gründe gegen eine naive Verwendung von *universal plans*, die auf situationsbasierte Arbeiten entsprechend übertragbar sind.

Zielrevision, also das Hinzufügen, Streichen, Verändern oder Gewichten von Planzielen anläßlich der Planung, ist im Kontext der KI-Planung noch nicht systematisch untersucht worden, obwohl es ein „natürliches“ Phänomen ist (Beispiele gibt beispielsweise [Dörner, 1989]) und in etlichen Planungsanwendungen auftaucht; zwei Beispiele sind die Constraint-Abschwächung in ISIS [Fox und Smith, 1984] und die nachträgliche Einführung von Vermeidungszielen in CHEF [Hammond, 1989]. [Hertzberg, 1990] gibt einen ersten, technisch und praktisch allerdings noch unbefriedigenden Einstieg in eine systematischere Betrachtung des Themas Zielrevision.

Eine Übersicht über neuere Planungsansätze, die die Themen ein wenig anders schneidet, gibt beispielsweise [McDermott, 1992]; weiterhin enthält die Planungs-Gesamtübersicht [Hendler *et al.*, 1990] auch Teile über nichtklassische Planungsansätze.

4.1 Unvollständigkeit der Information beim Planen

In vielen praktischen Anwendungsfällen ist zur Planungszeit nicht vollständig oder nicht sicher bekannt, in welchem Zustand sich die Welt bei Beginn der Planausführung befinden wird und welche Effekte einzelne Handlungen tatsächlich haben werden. Die Reaktion des klassischen Planens auf diese Tatsache wäre, für den plausibelsten Startzustand und die am wahrscheinlichsten zu vermutenden Handlungseffekte zu planen und neu- oder umzuplanen, wenn sich bei Planausführung herausstellt, daß die Welt anders ist. Damit mag man in Einzelfällen auskommen, aber im allgemeinen ist dieses Verfahren sehr unelegant und vermutlich ineffektiv.

Es gibt zahlreiche Arbeiten, diese Unsicherheit bereits zur Planungszeit in dem Sinne zu berücksichtigen, daß der Plan die bekannten oder wenigstens die berücksichtigenswerten Alternativen beachtet und Reaktionen auf ihr Auftreten enthält. Alle diese Arbeiten setzen allerdings pragmatisch immer noch voraus, daß die Unsicherheit über die Welt relativ gering ist: In einer Welt da im wesentlichen Unwissenheit und Zufall regieren, sollte man das Planen von vornherein seinlassen. (Zielgeleitet handeln kann man dort allenfalls „aus der Situation heraus“, wie oben angesprochen.)

Zwei Klassen von Techniken für diesen Fall sollen hier näher betrachtet werden: Erstellen von Konditionalplänen und ein entscheidungstheoretisch motivierter Planungsansatz. Der Aktionskalkül MW, der in Abschnitt 5 vorgestellt wird, berücksichtigt die angesprochene Art von Unsicherheit ebenfalls, und die in Abschnitt 6 skizzierte Art der auf MW aufsetzenden Planung ist stark durch Ideen aus der Entscheidungstheorie beeinflußt; diese beiden Abschnitte vertiefen folglich das Thema „Unvollständigkeit der Information beim

Planen" durch Beispiele.

Beginnen wir mit Konditionalplänen. Konditionalpläne sind der naheliegende Ansatz, wenn man – wie aus dem klassischen Planen übernommen – Pläne als eine Art von Computerprogrammen ansieht und Unvollständigkeit der Information über auftretende Situationen voraussetzt: Sie enthalten Verzweigungen von der Art **if**...**then**...**else if**...**then**... wie sie in Programmiersprachen üblich sind. Dazu enthalten Konditionalpläne boolesche Ausdrücke über Situationen, die bei Planausführung ausgewertet werden, um mit dem entsprechend passenden Plan-Ast weiterzumachen.

Da diese Idee so naheliegend ist, wird sie schon lange verfolgt, und zwar erstmalig von Green [1969]. Eine modernere Formulierung in nichtlinearen Plänen bringt [Peot und Smith, 1992]. Der ausdrückliche Zusammenhang mit der automatischen Generierung von Programmen wird in [Manna und Waldinger, 1987] hergestellt; der dort beschriebene Kalkül erlaubt allerdings nicht nur bedingte Verzweigungen, sondern auch beispielsweise **while**-Schleifen – eben die Erzeugung üblicher imperativer Programme.

Entscheidungstheoretische Planungsansätze verwenden ebenfalls Plan-Verzweigungen, bringen aber noch eine weitere Idee hinzu, die übrigens von Verzweigungen im Prinzip unabhängig ist. Es geht bei diesen Ansätzen nicht primär darum, Pläne zu erzeugen, die Ziele erfüllen, sondern solche, die den erwarteten Nutzen ihrer Ausführung maximieren. Erreichen der Ziele schlägt dabei mit hohem Nutzen zu Buche, aber auch Erreichen nur eines Teils der Ziele kann bereits Nutzen bringen. Lediglich *erwarteter* Nutzen kann im allgemeinen mit Plänen assoziiert werden, weil eben Unsicherheit über die Welt vorausgesetzt ist, wobei diese Unsicherheit durch Eintrittswahrscheinlichkeiten quantifiziert sein muß. Überdies haben Handlungen Kosten, die ebenfalls quantifiziert sein müssen.

Betrachten wir als Beispiel das *Putzproblem.* In einem Haushalt gebe es zwei anstehende Putzaufgaben: Fensterputzen und Abwaschen. Im Haushalt leben zwei hausarbeitsfähige Akteure: der Planer-Akteur PA und der Andere, A. Die Sauberkeit der Fenster hat einen hohen Wert von, sagen wir, 100; sauberes Geschirr sei 50 wert. Dafür ist Fensterputzen auch lästiger: es kostet 50 Einheiten, Abwaschen nur 20 Einheiten, weshalb beide Akteure im Zweifel lieber abwaschen. PAs Bereichsrepräsentation enthält das Faktum, daß A erst dann mit irgendeiner Arbeit anfängt, wenn PA mit seiner schon fertig ist.[18] Eine weitere Vorbedingung aller Operatoren sei, daß die Arbeitsverteilung an PA und A insgesamt geklärt ist – vorher fängt auch PA nicht an.

Abbildung 7 enthält einen Plan aus Sicht von PA: Es wird um die Arbeitsverteilung eine Münze geworfen, und anschließend fängt PA mit seiner Aufgabe an. An den Kanten im Plan ist außer den erzeugten Nachbedingungen und deren Nutzen die Übergangswahrscheinlichkeit vermerkt. In den Kästchen, die die Operatoren darstellen, sind die Operatorkosten eingetragen; die Kosten dafür, daß A arbeitet, sind aus Sicht von PA jeweils 0. Die Kosten des Münzwurfs legen wir im Moment nicht fest, sondern nennen sie c_M.

Ein Plan, dessen Ausführbarkeit aus Sicht von PA gewährleistet ist, ist der mit der Arbeitsverteilung, daß PA alles selber macht. Wenn wir Kosten als negativen Nutzen definieren, ist der Nutzen dieses Plans klar: die Nutzen-Summe minus die Kosten-Summe, also $150 - 70 = 80$. Der Nutzen des Putz-Plans in Abbildung 7 ist weniger sicher, da nicht klar ist, welcher Plan-Ast ausgeführt werden wird. Man hilft sich deshalb mit dem

[18] As Bereichsrepräsentation enthält die genau umgekehrte Information. Das ist ein Beispiel von Inkonsistenz des Weltwissens beim verteilten Planen und ein empirisches Phänomen in entsprechenden Haushalten.

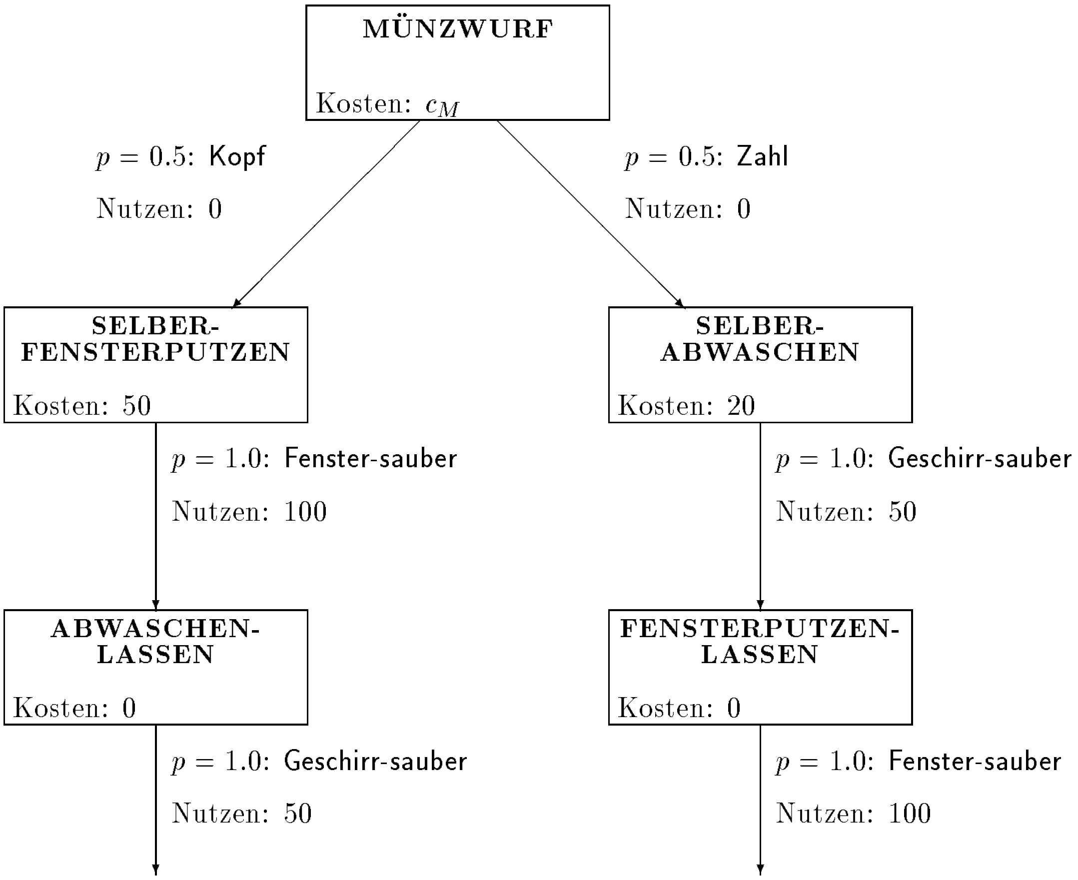


Abbildung 7: Plan zur Lösung des Putzproblems. Erklärungen im Text.

erwarteten Nutzen, der für einen Plan Π mit Ereignissen Ω (Ereignisse sind Operatoren und das Eintreten von Bedingungen) so definiert ist:

$$U(\Pi) = \sum_{\omega \in \Omega} U(\omega) p(\omega \mid \Pi) \tag{1}$$

Dabei gibt die Funktion $U(\omega)$ („*utility*") den Nutzen des Ereignisses ω wieder. Für den Putz-Plan ergibt sich also als erwarteter Nutzen:

$$-c_M \cdot 1 + (0 - 50 + 100 - 0 + 50) \cdot 0.5 + (0 - 20 + 50 - 0 + 100) \cdot 0.5 = 115 - c_M$$

Das bedeutet: Sollte der Münzwurf mehr als 35 kosten (beispielsweise weil damit gruppendynamische Diskussionen über die Fairness von Arbeitsverteilung verbunden sind), sollte PA lieber alles selbermachen; kostet es mehr als 20, also mehr als die Nutzen-Differenz zwischen dem ungünstigeren Ast und dem Plan, alles selberzumachen, dann läuft PA gefahr, damit schlechter wegzukommen als mit dem Selbermach-Plan. Wenn schließlich PA beim Münzwurf so schummeln kann, daß er in 70% aller Fälle Zahl wirft, ändert sich der erwartete Nutzen des Putz-Plans zu $121 - c_M$.

Das Beispiel zeigt, daß der entscheidungstheoretische Ansatz im wesentlichen die Definition eines rational begründeten Qualitätsmaßes zu der Behandlung von Verzweigungen hinzufügt. Es erleichtert zunächst nicht, einen Plan oder auch nur Äste eines Plans zu *finden*. Allerdings kann ein gutes Qualitätsmaß natürlich erheblich dabei helfen, beispielsweise heuristische Suchverfahren zum Planen einzusetzen.

Bereits 1977 wurde in [Feldman und Sproull, 1977] für die Verbindung von Planen mit Ideen aus der Entscheidungstheorie plädiert. Diese Arbeit geriet dann aber eine Zeitlang in Vergessenheit, bis Ende der 80er Jahre verstärkt über Planen unter Unsicherheit geforscht wurde. Eine sehr gute Einführung in das Thema enthält [Dean und Wellman, 1991, Kap. 7].

4.2 Rechenzeitbeschränkungen beim Planen

Es gibt Problembereiche, in denen von der Rechenzeit, die für die Erstellung eines Plans gebraucht wird, nicht abstrahiert werden darf. Standard-Literaturbeispiel ist der Roboter, der die Straße überquert und einen Lastwagen auf sich zurasen sieht: Es mag unterschiedlich gute Reaktionen in dieser Situation geben, aber am allerwichtigsten ist, *schnell irgendetwas* zu tun, selbst wenn sich später bei ruhiger Überlegung herausstellen sollte, daß es eine bessere Reaktion gegeben hätte – daß beispielsweise vorzuspringen statt zurückzuspringen den insgesamt zurückzulegenden Weg um den Betrag der Straßenbreite verkürzt hätte. Der Merksatz für solche Anwendungsgebiete ist:

> Ein suboptimaler oder gar nur halbfertiger Plan zur rechten Zeit ist nützlicher als ein optimaler zu spät!

Letztlich liegt diese Haltung den „zufriedenstellenden" (*satisficing*) Lösungen zugrunde, die laut Simon [1976] Individuen mit ihrer normalerweise begrenzten Rationalität (*bounded rationality*) erreichen können.

An dem Merksatz lassen sich zwei Teilprobleme erkennen, die beim rechenzeitbeschränkten Planen auftreten:

1. Wie kann man gewährleisten, daß der Planer „zur rechten Zeit" einen Plan zur Verfügung hat – und sei er auch nur suboptimal? Schließlich hatten wir beim klassischen Planen gesehen, daß Planen beliebig komplex sein kann.
2. Wie definiert man ein Maß der Nützlichkeit von Plänen, das die verbrauchte Rechenzeit einbezieht?

Beginnen wir mit dem zweiten Punkt, weil er unmittelbar an die entscheidungstheoretischen Planungsmodelle des vorigen Abschnitts anschließen kann. Die Grundidee ist, den erwarteten Nutzen des Weiter*planens* zu vergleichen mit dem erwarteten Nutzen, der aus unmittelbarer Ausführung des vorhandenen (möglicherweise noch unfertigen) Plans resultiert. Letzteren Nutzen setzen wir für einen Plan Π vereinfachend gleich mit dem in (1) definierten $U(\Pi)$. Der erwartete Nutzen des Weiterplanens kann natürlich nur geschätzt werden. Gebe $U^*(\Pi, t', t)$ die Schätzung des erwarteten Nutzens desjenigen Plans, der sich ergibt, wenn der Plan Π, für dessen Erstellung bereits t' Zeit verbraucht wurde, noch t Zeit weiter entwickelt wird. Dann ist der erwartete Nutzenzuwachs

$$\Delta_U(\Pi, t', t) = U^*(\Pi, t', t) - U(\Pi)$$

Diesen Wert vergleicht man mit den Kosten $C(t)$ von t Zeiteinheiten Planung: Weiterplanen macht Sinn, wenn $\Delta_U(\Pi, t', t) > C(t)$

Dieses Berechnungsmodell ist natürlich stark vereinfacht. Zum Beispiel können die Planungskosten situationsabhängig drastisch schwanken: Für den Roboter im Straßenverkehr können 10 Sekunden Planungszeit deutlich teurer sein als nach Feierabend im häuslichen Maschinenschuppen. Weiterhin ist nicht klar, ob die Berechnung des erwarteten Nutzens eines möglicherweise unfertigen Plans sinnvoll mit $U(\Pi)$ berechnet werden darf. Es geht hier jedoch nur darum, das Prinzip zu erklären. [Russell und Wefald, 1991] befassen sich ausführlich mit diesem Problem.

Die andere eben angesprochene Frage war die, wie gewährleistet werden kann, daß ein Planer jederzeit einen sinnvollen Plan oder wenigstens einen sinnvollen als nächsten auszuführenden Operator zur Verfügung stellen kann. Bei der Prozedur NLP in Abschnitt 3 ist das offenbar nicht der Fall: In der Zeit, die sie läuft, ist eben *nicht* gewährleistet, daß ein nächster, sinnvoll auszuführender Schritt bekannt ist. Es liegt in der Natur der Sache, daß ein solcher vorzeitig zurückgegebener Operator nicht in irgendeinem Sinn optimal sein muß: Es kann immer sein, daß man noch einen besseren gefunden hätte, hätte man länger planen dürfen.

Nach [Dean und Boddy, 1988] bezeichnet man Planer, die im Prinzip jederzeit einen nächsten auszuführenden Operator zurückgeben können, als *anytime*-Planer. Bei näherem Hinsehen [Russell und Zilberstein, 1991] stellt man fest, daß es mindestens zwei Varianten von *anytime*-Planern gibt:

unterbrechbare (*interruptible*) Planer, das sind solche, die zu jedem Zeitpunkt ihrer Laufzeit unterbrochen werden können; und

terminierbare (*contract*) Planer, das sind solche, denen man vor Beginn ihres Laufs einen Termin setzen kann, bis zu dem sie einen Plan zurückgegeben haben müssen.

Beide Versionen von *anytime*-Planen sind in einem streng formalen Sinn insofern trivial, als praktisch jede Planungsprozedur trivial zu einer *anytime*-Prozedur modifiziert werden kann: Man nehme eine globale Variable, der anfangs ein Standardplan zugewiesen wird (z. B. „Warte“); wenn die Prozedur fertig ist, weise man der Variablen das Ergebnis zu; wird sie nicht fertig, gebe man den Standardplan zurück.

„*Anytime* oder nicht“ ist also eher ein intuitiv begründbares Kriterium, das sich gemessen am Sinn des Planens im Anwendungsbereich ergibt. Ein „sinnvoller“ *anytime*-Planer sollte Anforderungen wie die folgenden erfüllen:

- Je mehr Zeit der Planer hat, desto besser (bezüglich eines Plan-Qualitätsmaßes) sollen im Schnitt seine Ergebnisse werden.
- Der Planer bearbeitet diejenigen offenen Planteile zuerst, deren Komplettierung die Qualität des Plans am meisten zu verbessern verspricht.
- Der Plan, den ein unterbrechbarer Planer zurückgibt, soll wenigstens in einem Anfangsstück ausführbar sein.
- Die Information über verbleibende Restlaufzeit eines terminierbaren Planers soll dessen Laufzeitverhalten verändern. („Packe nur noch Planteile an, mit denen Du voraussichtlich fertig wirst!“)

Auch zum Umgang mit Rechenzeitbeschränkungen enthält [Dean und Wellman, 1991] (besonders Kapitel 8) eine gute Einführung. In Abschnitt 6 werden wir eine Version von Rechenzeitbeschränkung formal beschreiben und verwenden.

4.3 Dynamik und Unbekanntheit der Welt beim Ausführen

Die Alltagserfahrung lehrt, daß die Ausführung von Plänen scheitern oder stocken kann, und oft liegt das Problem dann darin, daß die Welt sich nicht so verhält wie geplant. Man kann das als Mangel an Information zur Planungszeit interpretieren, aber spätestens Effektivitätserwägungen setzen dem Detaillierungsgrad der Bereichsrepräsentation für den Planer Grenzen (s. Abschnitt 2.1), so daß man irgendwann der Tatsache ins Auge sieht: Bei der Planausführung kann Unerwartetes geschehen.

Wiederum – wie bei der Unsicherheit zur Planungszeit – wäre eine denkbare Haltung die, das Unvorhersehbare eintreten zu lassen und daraufhin neu zu planen. Aber wiederum kann man auch versuchen, geschickter vorzugehen.

Einmal mehr stammt der erste Ansatz dazu [Fikes *et al.*, 1972] aus dem Kontext von STRIPS. Ihm lag die Beobachtung zugrunde, daß vieles Unvorhergesehene einfach darauf beruht, daß Handlungen oder Planteile ungenau ausgeführt werden, also einzelne oder alle ihrer normalen Nachbedingungen nicht erzeugen, dafür aber möglicherweise andere. Damit versetzen sie aber die Welt nicht in einen völlig unbekannten Zustand weitab von allem Geplanten, sondern mitunter müssen sie einfach wiederholt werden, damit alles nach Plan weiterlaufen kann. Um solche Fälle abzufangen, wurde für PLANEX, das Planausführungsmodul von STRIPS, eine alternative Darstellungsform für lineare Pläne entwickelt: die *Dreieckstafel*.[19]

Eine Dreieckstafel ist im wesentlichen eine kompakte Darstellung der Abhängigkeitsstruktur in einer linearen Operatorfolge. Abbildung 8 zeigt als Beispiel die Dreieckstafel für den Plan aus Abbildung 5 zur Lösung der Sussman-Anomalie. Am Ende jeder i-ten

[19] Die Dreieckstafel ist auch das Format, in dem STRIPS-Pläne zur Wiederverwendung als Makro-Operatoren dargestellt werden, wie weiter oben angesprochen.

0 $\mathcal{I}$							
Hand(NIL) Frei(C) Auf(C, A)	1 **TRENNE**(C, A)						
	Hand(C)	2 **SETZE**(C)					
Tisch(B) Frei(B)		Hand(NIL)	3 **NIMM**(B)				
		Frei(C)	Hand(B)	4 **STAPLE**(B, C)			
Tisch(A)	Frei(A)			Hand(NIL)	5 **NIMM**(A)		
				Frei(B)	Hand(A)	6 **STAPLE**(A, B)	
				Auf(B, C)		Auf(A, B)	$\mathcal{G}$

Abbildung 8: Dreieckstafel für den Plan zur Lösung der Sussman-Anomalie; der 4. Kern ist markiert. Die Ziffern geben die Nummer der Spalte an, *über* der, und die Nummer der Zeile, *neben* der sie stehen. Weitere Erklärungen im Text.

Zeile der Tafel steht der i-te anzuwendende Operator, und in Zelle (i, j) stehen diejenigen Merkmale, die der j-te Operator als Vorbedingungen für den i-ten erzeugt. Der i-te *Kern* einer Dreieckstafel ist das maximal breite Rechteck, das die Zelle $(i, 0)$ enthält; er enthält alle Merkmale, die gelten müssen, damit der gesamte Restplan ab dem i-ten Operator ausgeführt werden kann. Sind alle Merkmale eines Kerns erfüllt, heißt er ebenfalls *erfüllt.*

Eine Dreiecktafel wird nun folgendermaßen ausgeführt.

1. Bestimme den erfüllten Kern mit dem aktuell höchsten Index; sei dieser Index i.
2. Falls i die Zeilenzahl der Dreieckstafel ist, ist die Ausführung fertig.
3. Ansonsten führe den i-ten Operator aus und mache weiter bei (1).

Läuft die Planausführung wie vorgesehen, wird nach diesem Verfahren genau der vorliegende Plan abgearbeitet. Um zu sehen, was passiert, wenn Ausführungsfehler auftreten betrachten wir drei Beispiele:

- Operator 3 (**NIMM**(B)) möge so scheitern, daß er nichts verändert („danebengegriffen"). Danach ist der 3. Kern der höchste erfüllte, und **NIMM**(B) wird direkt wieder ausgeführt.
- Operator 1 (**TRENNE**(C, A)) möge so scheitern, daß C auf dem Tisch liegt („aus der Hand gerutscht"). Danach ist der 3. Kern der höchste erfüllte, und der Operator **SETZE**(C) wird einfach übersprungen.
- Operator 4 (**STAPLE**(B, C)) möge so scheitern, daß B auf dem Tisch liegt („runtergefallen"). Wiederum ist dann der 3. Kern der höchste erfüllte, und B wird nochmal aufgehoben.

Andere, neuere Ansätze, die auf einer vergleichbaren Transformation von Plänen in ein anderes Ausführungsformat beruhen, sind beispielsweise *Reaktionspläne* (z.B. Georgeff und Lanskys [1987] PRS) oder *situated control rules* [Drummond, 1989].

Diese Art der Planausführung macht jedoch eine entscheidende Voraussetzung: Die Information über das Zutreffen bzw. Nicht-Zutreffen von Fakten ist mit vernachlässigbarem Aufwand erhältlich. Das muß aber nicht der Fall sein. Ein Beispiel aus der Lebenswelt ist der Gebrauchtwagenkauf: Der technische Zustand eines Gebrauchtwagens ist für einen Laien nur schwer abschätzbar; man kann ihn durch einen Gutachter überprüfen lassen – das verursacht aber Kosten in D-Mark. Bei der Ausführung von Plänen durch autonome Roboter entstehen praktisch immer Kosten in Rechenzeit dafür, Merkmale in der Welt zu prüfen, da die Robotersensorik normalerweise keine Prädikate messen kann, sondern beispielsweise boolesche oder numerische Werte, die erst aggregiert und in die symbolische Bereichsrepräsentation umgerechnet werden müssen. Auf dieses im allgemeinen schwierige Problem gehe ich hier nicht weiter ein; [Nilsson, 1984] vermittelt einen Eindruck davon, wie diese Probleme in STRIPS/Shakey behandelt wurden.

Wenn Informationen über die Welt zur Ausführungszeit nicht vernachlässigbar billig sind, ist nicht nur das wiederholte Prüfen der Erfülltheit von Kernen in Dreieckstafeln zu teuer, sondern es wird auch wird die Verwendung von Konditionalplänen in Frage gestellt. Literaturbeispiel dafür ist das *Briefbombenproblem:*

> Es liegen zwei Päckchen P, Q vor; eines davon enthält eine Wertsendung, das andere eine Briefbombe; Briefbomben entschärft man, indem man sie in einen Wassereimer steckt. Wie kommt man an die Wertsendung?

Ein naheliegender Konditionalplan wäre der: „Wenn die Bombe in P ist, öffne Q, andernfalls öffne P." Genau diese Bedingung kann aber nur mit unter Umständen hohen Kosten nachgeprüft werden.

Ein vernünftiger Ansatz in diesem Fall ist, *Sensoroperatoren* zu formulieren, die bereits der Planer in den Plan einbauen kann. Ein Sensoroperator hat als Nachbedingung die Information über Fakten in der Welt. Wie jeder Operator wird er normalerweise nicht zur Planungszeit ausgeführt, sondern was bei der Planung berücksichtigt werden kann, ist zum einen, daß nach Ausführung solch eines Operators die Wahrheit oder Falschheit des fraglichen Faktums bekannt sein wird, und zum zweiten die Kosten der Messung: Vielleicht gibt es Möglichkeiten, einen Plan zu finden, ohne mit der nur teuer meßbaren Information arbeiten zu müssen, oder es ist klar, daß die Kosten der unvermeidlichen Messung den Nutzen des Plans überwiegen werden, oder das nicht sicher bekannte Merkmal hat einen zuverlässig verwendbaren *default*-Wert („Die Gebrauchtwagen von Auto-Meier waren immer in Ordnung"). Enthält ein Plan allerdings allzuviele Sensoroperatoren, insbesondere viele billige, dann erscheint es sinnvoll, Planen und Planausführung zu verzahnen in dem Sinn: Führe soweit aus, wie Du kommst, miß die Ausprägung der dann angegebenen Merkmale, und plane entsprechend weiter. Neuere Arbeiten zur Verzahnung von Planen und Ausführen stammen beispielsweise von Firby (z.B. [Firby, 1992]); eine Arbeit zum Verwenden von Merkmal-*defaults* ist [Krebsbach *et al.*, 1992].

5 Aktionskalküle

In den Abschnitten 3 und 4 konnte man sehen, daß Planen die Fähigkeit zum Erschließen von Wandel voraussetzt: beispielsweise immer wenn berechnet werden soll, ob ein Merkmal an einer Stelle des Plans bereits gilt, muß der Planer es können. Ja, man könnte sagen, Erschließen von Wandel ist fürs Planen eine genauso grundlegende Voraussetzung, wie etwa Deduktion für den Bau von deduktiven Datenbanken.

Die Sache hat einen Haken. In der klassischen Logik herrscht weitgehend Einigkeit darüber, ob eine Formel aus einer Menge anderer Formeln folgt. Um das alte Beispiel zu bemühen: wenn ich weiß, daß alle Menschen sterblich sind und daß Sokrates ein Mensch ist, ist der Schluß akzeptiert, daß Sokrates sterblich ist. Das Gebiet Deduktion[20] kann darauf aufbauen und Kalküle definieren und implementieren, die dieser Intuition entsprechen, also korrekt sind. Natürlich gibt es Fälle, in denen die klassische Logik nicht angemessen ist. In diesen Fällen wird man sie angemessen modifizieren, beispielsweise die Monotonie aufheben und nichtmonotone Logiken[21] definieren. Auch für diese Logiken gibt es eine recht gute Intuition darüber, ob eine Formel aus einer Formelmenge folgen soll – wenn auch vielleicht nicht mit so breiter Akzeptanz wie im klassischen Fall. Doch wiederum kann man korrekte Kalküle definieren, implementieren und benutzen.

Diese Intuition scheint beim Erschließen von Wandel nicht so gut entwickelt zu sein. Damit zusammenhängend ist theoretisch und technisch besonders störend: Es gibt keine formale Semantik von Operatoren, die der Semantik einer logischen Formel entspräche. Und ein weiteres Problem: Das Ergebnis der Anwendung eines Operators hängt stark ab

[20] Zur Übersicht s. beispielsweise [Walther, 1993].

[21] Zur Übersicht s. beispielsweise [Brewka, 1993].

von der Ausprägung der Merkmale des betrachteten Anwendungsbereichs (s. 2.2). Genauso, wie es nicht *das* Planen gibt, gibt es also auch nicht *das* Erschließen von Wandel.

Diese grundlegenden Probleme wurden in den meisten Arbeiten zum Planen außerordentlich pragmatisch behandelt: Sie wurden ignoriert. Man hat – plakativ formuliert – Planer gebaut wie in den vorangehenden Abschnitten beschrieben, hat zugesehen, daß die Pläne, die herauskamen, für die betrachteten Anwendungsbereiche oder Spielbeispiele halbwegs plausibel aussahen, und hat Fragen wie die Bedeutung eines Plans oder gar eine formale Semantik von Plänen großzügig übergangen.

Das kann man natürlich tun. Besonders als das Gebiet Planen noch relativ neu war und praktische Erfahrung mit Planern fehlte, war diese Strategie plausibel. Doch denke ich, inzwischen sollten wir mehr wollen. Will man nicht nur Pläne erzeugen, sondern auch noch sagen können, was sie *bedeuten,* muß man das Planen auf den formalen Grund des Erschließens von Wandel stellen, und darum geht es nun. Das soll *nicht* bedeuten, jeder Planer solle um einen korrekten und vollständigen Aktionskalkül „herumgebaut" werden: wir werden gleich sehen, daß solche Kalküle beliebig kompliziert und im allgemeinen sogar unentscheidbar werden können, so daß eine direkte Implementierung als Basis-Prozedur eines Planers mit praktikablen Laufzeiten sich selbst verbietet. In Abschnitt 6 wird ein sinnvolles Verhältnis von Kalkülen und praktikablen Planern ausführlicher besprochen. Hier soll zunächst *ein* solcher Kalkül ausführlicher vorgestellt werden.

Demnach geht es hier *nicht* darum, einen Überblick über Aktionskalküle zu geben. An dieser Stelle will ich aber wenigstens Hinweise auf andere als den im folgenden vorgestellten Kalkül geben. Urvater aller KI-Aktionskalküle ist der bereits in Abschnitt 2 angesprochene Situationskalkül von McCarthy und Hayes [1969]. Eine Familie von Aktionskalkülen, die unter anderem numerische Zeit einbeziehen, gibt [Sandewall, 1992] an, das auch weitere Referenzen auf Aktionskalküle insgesamt enthält. Dem im folgenden vorgestellten Kalkül am ähnlichsten sind andere Exemplare aus der Familie der sogenannten „formelbasierten" Kalküle wie [Ginsberg und Smith, 1988a; Winslett, 1988; Cordier und Siegel, 1992]. Der im folgenden beschriebene Kalkül ist der aus [Brewka und Hertzberg, 1993]; in diesem Text sind auch die Unterschiede zu den anderen formelbasierten Kalkülen detaillierter angegeben und begründet. Weitere Aktionskalküle sind diejenigen, die dem deduktiven Planen zugrundeliegen, s. [Biundo, 1994] für einen Überblick; Grundidee beim deduktiven Planen ist, einen Theorembeweiser für den entsprechenden Kalkül direkt als Planer einzusetzen. Warum ich das genau *nicht* machen will, wird in Abschnitt 6.3 begründet.

Die Formalisierung von Zustandsveränderungen ist außerhalb der KI oder des Planens ebenfalls Thema in den Bereichen Semantik, Verifikation oder automatische Generierung von (imperativen) Programmen. Über die Arbeit von Rosenschein [1981] ist von den entsprechenden Formalismen besonders die dynamische Logik im Planen als zugrundeliegender Formalismus wahrgenommen worden. Allerdings unterliegt die Transformation von Programmzuständen durch Zuweisungen anderen Randbedingungen als der Übergang von Situationen durch beliebige Handlungen, so daß zumindest nicht offensichtlich scheint, wie beliebige Aktionskalküle in dynamischer Logik reformuliert werden können.

5.1 MW: Ein Aktionskalkül mit möglichen Welten

Beginnen wir mit einer Zusammenfassung: Was soll der nun zu beschreibende Kalkül MW eigentlich berechnen? Ausgangspunkt ist, daß ein Zustand der Welt durch eine Menge beliebiger prädikatenlogischer Formeln beschrieben ist, also nicht nur durch Grundatome wie im klassischen Planen. Seien Operatoren wie gehabt beschrieben durch ihre Vor- und Nachbedingungen, die allerdings ebenfalls beliebige prädikatenlogische Formeln sein dürfen sollen. Dann soll MW für eine Situation (also eine Formelmenge) und einen Operator angeben, welche Folgesituation entsteht bzw. welche Folgesituation*en* entstehen können, falls das beispielsweise aufgrund von Schneeballeffekten nicht eindeutig ist. Als Spezialität soll MW erlauben, daß die Effekte von Handlungen kontextabhängig sind, also in unterschiedlichen Zuständen unterschiedlich ausfallen können, und daß im selben Zustand Alternativresultate herauskommen können wie beispielsweise beim Werfen einer Münze, wo entweder Kopf oder Zahl herauskommt, man weiß aber vorher nicht, was. In den Begriffen von Abschnitt 2.2 ist MW also gedacht für Anwendungsbereiche mit den folgenden Charakteristika:

- Zeitmodell: qualitativ, situationsbasiert
- Bereichsdarstellung: möglicherweise unvollständig
- Aktionen: möglicherweise kontextabhängig
- Aktionseffekte: möglicherweise nicht-deterministisch

Über die anderen Bereichseigenschaften aus 2.2 setzt MW nichts voraus; wir werden später beispielsweise sehen, daß Pläne, die auf ihm beruhen, robust sein können gegenüber dynamischen Veränderungen im Bereich.

Die folgenden Prinzipien liegen nun MW und anderen, vergleichbaren Kalkülen (den „formelbasierten" Kalkülen) zugrunde:

- Zustände werden beschrieben durch Mengen prädikatenlogischer Formeln, genannt Situationen.
- Es gibt eine Formelmenge K („*knowledge*", Hintergrundwissen), die solche Zusammenhänge beschreibt, die sich durch keinen Operator jemals ändern kann und die in jeder Situation gelten muß. Das sind also beispielsweise Bereichsaxiome oder Definitionen; Beispiele solcher Formeln in der Blockwelt sind

$$\forall x, y, z.[\mathsf{Auf}(x, z) \land x \neq y \rightarrow \neg\mathsf{Auf}(y, z)]$$

$$\forall x.[\neg\exists y.\mathsf{Auf}(y, x) \rightarrow \mathsf{Frei}(x)]$$

- Eine Nachbedingung $Post(o)$ eines Operators o ist beschrieben durch eine Formelmenge, die in der Nachfolgesituation gültig zu sein hat. Es gibt also keine Unterscheidung zwischen hinzuzufügenden und zu löschenden Formeln.
- Es bestehen die folgenden Anforderungen an eine Situation S', die durch Anwendung des Operators o in S entstehen, sofern er anwendbar ist:[22]

[22] Der flüssigen Lesbarkeit halber bezeichne ich mit S, S' im folgenden sowohl Formelmengen als auch deren Modelle. Im Kontext sollte jeweils klar sein, was gemeint ist.

1. $S' \models Post(o)$, d.h. die Nachbedingungen gelten in S',
2. S' ist eine maximale mit K konsistente Formelmenge, d.h. die Nach-Situation läßt nichts unbestimmt, und
3. S' ist minimal unterschiedlich von S, d.h. es ändert sich nur das, was durch $Post(o)$ erzwungen ist.

Das entscheidende Problem ist, Forderung (3) einzulösen, die offensichtlich das Rahmenproblem lösen soll. Es scheint keine klare, allgemein verbreitete Intuition darüber zu geben, was es heißt, daß zwei Formelmengen minimal unterschiedlich sind.

MW verwendet zur Bestimmung der Frage, was eine minimal unterschiedliche Formelmenge ist, das Konzept „Kausalität"[23] Wenn eine geänderte Formel φ als Folgeänderungen die Formeln $\varphi_1, \ldots, \varphi_n$ „kausal" nach sich zieht, dann soll das nur als 1 Änderung zählen, denn φ „bedingt" $\varphi_1, \ldots, \varphi_n$. Der Nebeneffekt dabei ist: Ändert man eine Situation so, daß sie nun φ enthält, *muß* man akzeptieren, daß in der neuen Situation $\varphi_1, \ldots, \varphi_n$ gelten.

Basis von MW ist eine endliche logische Sprache $\mathcal{L}$ erster Ordnung, also beispielsweise eine Sprache, die in Prädikatenlogik erster Stufe die Prädikatensymbole und Konstanten enthält, um die Blockwelt zu beschreiben. Alle Formeln, von denen im folgenden die Rede ist, sind im Zweifel Formeln aus $\mathcal{L}$. Jede maximale konsistente Konjunktion von Grundliteralen aus $\mathcal{L}$ heißt eine *Welt.*[24]

Als Beispiel-Anwendungsbereich taugt die Blockwelt aber nun nicht mehr, weil es in ihr beispielsweise keine kontextabhängigen Operatoreffekte gibt. Um die nun vorzustellenden Konzepte zu verdeutlichen, benutze ich die *Tassenwelt,* die ungefähr genauso künstlich ist; Abbildung 9 soll die Vorstellung beflügeln. Der Einfachheit halber ist die Tassenwelt folgendermaßen *aussagelogisch* beschrieben. Aufgabe eines Roboters ist es, eine Tasse zu manipulieren. Die Tasse kann entweder auf dem Boden stehen (Boden) oder auf dem Tisch (Tisch). Auf dem Boden kann sie entweder aufrecht stehen ($\mathsf{Aufrecht}$) oder aus Sicht des Roboters vorwärts ($\mathsf{Vorwärts}$) oder rückwärts gekippt liegen ($\mathsf{Rückwärts}$). Die Arbeit findet unter freiem Himmel statt; Regenwetter (Regen) kann also die Arbeit beeinträchtigen, beispielsweise die Tasse rutschig machen. Die Sprache $\mathcal{L}_c$ der Tassenwelt ist die Sprache erster Ordnung, die durch die Grundatome $\{\mathsf{Boden}, \mathsf{Tisch}, \mathsf{Aufrecht}, \mathsf{Vorwärts}, \mathsf{Rückwärts}, \mathsf{Regen}\}$ induziert wird. Eine Welt in der Tassenwelt ist jede Menge von Grundliteralen, die aus genau diesen sechs Grundatomen entstehen.

Eine Situation ist eine Formel aus $\mathcal{L}$. Eine Situation muß einen Zustand der Tassenwelt nicht eindeutig beschreiben; ein Beispiel für eine solche mehrdeutige Zustandsbeschreibung ist

$$s_c = \mathsf{Regen} \wedge (\mathsf{Tisch} \vee \mathsf{Vorwärts})$$

[23]Mit demselben Argument, das mich im klassischen Planen vor der Verwendung des intuitiven Begriffs Abstraktion zurückschrecken ließ, könnte man hier den Begriff Kausalität *nicht* verwenden wollen, obwohl er in der Originalarbeit [Brewka und Hertzberg, 1993] verwendet wird: Es scheint eher fraglich, ob hier die intuitive Kausalität erfaßt ist. Ich bin inkonsequent und verwende den Begriff trotzdem.

[24]Eine Welt entspricht also syntaktisch einem Herbrand-Modell. Man beachte, daß die Begriffe „Welt" und gleich „mögliche Welt" Begriffe auf der *Syntax*ebene sind! Sie haben also nichts zu tun mit den möglichen Welten in der Semantik der Modallogik. Die Begriffsverwirrung stammt von [Ginsberg und Smith, 1988a], und ich schreibe sie fort.

Zur Notation: Ich verwende zur Abkürzung im folgenden Welten als *Mengen* von Literalen durcheinander mit Welten als *Konjunktionen* von Literalen. Im Kontext ist entweder unerheblich oder eindeutig, welche Form gemeint ist.

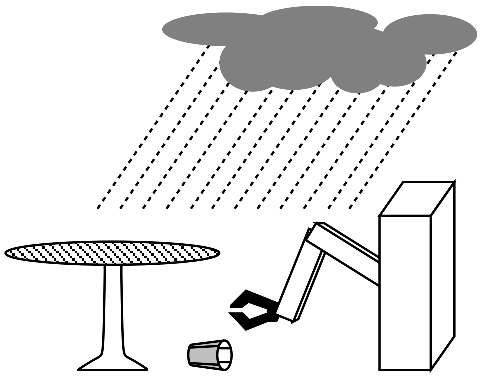


Abbildung 9: Eine bildliche Darstellung der Tassenwelt.

Das Hintergrundwissen liegt in zwei Teilen vor:

B, das *logische Hintergrundwissen:* Eine Menge von Formeln, die in allen Zuständen des Anwendungsbereichs gelten.

C, das *kausale Hintergrundwissen:* Eine Menge von Inferenzregeln der Form $\varphi \Rightarrow \varrho$ über $\mathcal{L}$.

In C sind Inferenzregeln statt Implikationen verwendet, um die Anwendung in Kontraposition zu verhindern: Aus einer Implikation $\varphi \rightarrow \varrho$ und $\neg\varrho$ kann $\neg\varphi$ abgeleitet werden, aus einer Inferenzregel $\varphi \Rightarrow \varrho$ und $\neg\varrho$ aber nicht. Als Notation verwende ich

$$\text{Theory}(C) = \{\varphi \rightarrow \varrho \mid \varphi \Rightarrow \varrho \in C\},$$

wobei freie Objektvariablen als allquantifiziert interpretiert werden. Theory(C) ist also die Menge aller Implikationen, die den Inferenzregeln in C entsprechen. Dann verwende ich als Abkürzung für das gesamte Hintergrundwissen

$$K = \text{Theory}(C) \cup B$$

In der Tassenwelt mögen die Merkmale Aufrecht, Vorwärts und Rückwärts wechselseitig widersprüchlich sein. Eines von ihnen bewirken, heißt, daß ebenfalls Boden gilt (egal, ob es vorher schon galt, oder nicht). Sind oder werden Aufrecht, Vorwärts und Rückwärts allesamt falsch, so bedeutet das, daß Tisch gelten muß. Alles das wird im kausalen Hintergrundwissen ausgedrückt. Schließlich definiere ich im logischen Hintergrundwissen Boden als das Gegenteil von Tisch, schließen also dritte Lagemöglichkeiten für die Tasse aus. B_c, C_c, und K_c sind dann folgendermaßen definiert:

$$B_c = \left\{ \mathsf{Boden} \leftrightarrow \neg\mathsf{Tisch} \right\}$$

$$C_c = \left\{ \begin{array}{c} \mathsf{Aufrecht} \Rightarrow \neg\mathsf{Vorwärts} \wedge \neg\mathsf{Rückwärts} \wedge \mathsf{Boden}, \\ \mathsf{Vorwärts} \Rightarrow \neg\mathsf{Aufrecht} \wedge \neg\mathsf{Rückwärts} \wedge \mathsf{Boden}, \\ \mathsf{Rückwärts} \Rightarrow \neg\mathsf{Vorwärts} \wedge \neg\mathsf{Aufrecht} \wedge \mathsf{Boden}, \\ \neg\mathsf{Aufrecht} \wedge \neg\mathsf{Vorwärts} \wedge \neg\mathsf{Rückwärts} \Rightarrow \neg\mathsf{Boden} \end{array} \right\}$$

$$K_c = \left\{ \begin{array}{c} \mathsf{Tisch} \leftrightarrow \neg\mathsf{Boden}, \\ \mathsf{Aufrecht} \rightarrow \neg\mathsf{Vorwärts} \wedge \neg\mathsf{Rückwärts} \wedge \mathsf{Boden}, \\ \mathsf{Vorwärts} \rightarrow \neg\mathsf{Aufrecht} \wedge \neg\mathsf{Rückwärts} \wedge \mathsf{Boden}, \\ \mathsf{Rückwärts} \rightarrow \neg\mathsf{Vorwärts} \wedge \neg\mathsf{Aufrecht} \wedge \mathsf{Boden}, \\ \neg\mathsf{Aufrecht} \wedge \neg\mathsf{Vorwärts} \wedge \neg\mathsf{Rückwärts} \rightarrow \neg\mathsf{Boden} \end{array} \right\}$$

Von allen Welten brauchen wir nur die *möglichen* Welten zu betrachten, also diejenigen, die mit dem Hintergrundwissen konsistent sind. Dazu definiere ich

Definition 10 (Mögliche Welt in s) *Sei s eine Formel, K das Hintergrundwissen. Die möglichen Welten in s bezüglich K sind*

$$\mathrm{Poss}_K(s) = \{w \mid w \textit{ ist eine Welt, und } K \cup \{s\} \not\models \neg w\}$$

Poss_K verwende ich als Abkürzung für $\mathrm{Poss}_K(\mathit{true})$, also die Menge aller Welten, die bezüglich K alleine möglich sind.

Für die Tassenwelt mit der oben angegebenen Situation s_c gilt dann

$$\mathrm{Poss}_{K_c}(s_c) = \left\{ \begin{array}{lcl} W_1 & = & \{\mathsf{Regen}, \neg\mathsf{Vorwärts}, \neg\mathsf{Rückwärts}, \neg\mathsf{Aufrecht}, \neg\mathsf{Boden}, \mathsf{Tisch}\}, \\ W_2 & = & \{\mathsf{Regen}, \mathsf{Vorwärts}, \neg\mathsf{Rückwärts}, \neg\mathsf{Aufrecht}, \mathsf{Boden}, \neg\mathsf{Tisch}\} \end{array} \right\}$$

(Die Namen W_i brauche ich später.) W_1, W_2 sind natürlich auch Elemente von Poss_{K_c}, das außerdem noch die folgenden Welten enthält:

$$\begin{array}{lcl} W_3 & = & \{\mathsf{Regen}, \neg\mathsf{Vorwärts}, \mathsf{Rückwärts}, \neg\mathsf{Aufrecht}, \mathsf{Boden}, \neg\mathsf{Tisch}\}, \\ W_4 & = & \{\mathsf{Regen}, \neg\mathsf{Vorwärts}, \neg\mathsf{Rückwärts}, \mathsf{Aufrecht}, \mathsf{Boden}, \neg\mathsf{Tisch}\}, \\ W_5 & = & \{\neg\mathsf{Regen}, \neg\mathsf{Vorwärts}, \neg\mathsf{Rückwärts}, \neg\mathsf{Aufrecht}, \neg\mathsf{Boden}, \mathsf{Tisch}\}, \\ W_6 & = & \{\neg\mathsf{Regen}, \mathsf{Vorwärts}, \neg\mathsf{Rückwärts}, \neg\mathsf{Aufrecht}, \mathsf{Boden}, \neg\mathsf{Tisch}\}, \\ W_7 & = & \{\neg\mathsf{Regen}, \neg\mathsf{Vorwärts}, \mathsf{Rückwärts}, \neg\mathsf{Aufrecht}, \mathsf{Boden}, \neg\mathsf{Tisch}\}, \\ W_8 & = & \{\neg\mathsf{Regen}, \neg\mathsf{Vorwärts}, \neg\mathsf{Rückwärts}, \mathsf{Aufrecht}, \mathsf{Boden}, \neg\mathsf{Tisch}\} \end{array}$$

Wir kommen nun zu der oben gestellten Frage, wie minimale Unterschiedlichkeit von Situationen – hier also von (möglichen) Welten definiert werden kann. Dazu dienen die nächsten beiden Definitionen, für die ich vorher noch ein wenig Notation brauche.

Für mögliche Welten w_1, w_2 bezeichnet $\mathrm{Diff}(w_1, w_2)$ die Menge der Grundliterale, die in w_2 wahr sind, aber nicht in w_1. In der Tassenwelt gilt also beispielsweise

$$\begin{array}{lcl} \mathrm{Diff}(W_1, W_2) & = & \{\mathsf{Vorwärts}, \mathsf{Boden}, \neg\mathsf{Tisch}\} \quad \text{und} \\ \mathrm{Diff}(W_2, W_1) & = & \{\neg\mathsf{Vorwärts}, \neg\mathsf{Boden}, \mathsf{Tisch}\} \end{array}$$

Sei I eine Menge von Inferenzregeln und F eine Formelmenge. Dann sei der *I-Abschluß* (notiert $\vdash_I$) von F die kleinste Formelmenge, die unter „normaler" Deduktion und Deduktion mit den Inferenzregeln aus I abgeschlossen ist. Ich schreibe $F \vdash_I \varphi$, wenn der I-Abschluß von F die Formel φ enthält. Ein Beispiel aus der Tassenwelt: der C_c-Abschluß der Formelmenge $\{\mathsf{Vorwärts}\}$ enthält $\neg\mathsf{Aufrecht} \wedge \neg\mathsf{Rückwärts} \wedge \mathsf{Boden}$ – der entsprechenden Regel aus C_c gehorchend; folglich enthält er auch $\neg\mathsf{Aufrecht}$, $\neg\mathsf{Rückwärts}$ und Boden; Regen ist nicht enthalten.

Wir kommen nun dazu, zu definieren, was es heißen soll, daß sich zwei Welten möglichst wenig unterscheiden, und ich brauche dazu die zwei folgenden, eher technischen Definitionen. Ich will ausdrücken, eine Welt w_1 liegt näher an der „Referenzwelt" w als w_2 es tut, wenn sich w_1 weniger als w_2 von w unterscheidet; die Rolle der „Referenzwelt" wird nachher von der Ausgangswelt einer Operatoranwendung übernommen.

Grundlage dafür, die Nähe von möglichen Welten zu definieren, ist die nun zu definierende *Kausalbasis.* Das ist eine minimale Teilmenge der mengentheoretischen Differenz, aus der durch C-Abschluß alle weiteren Komponenten der Differenz folgen. Formal:

Definition 11 (Kausalbasis) *C und B seien das kausale bzw. das logische Hintergrundwissen, w_1 und w_2 seien mögliche Welten, und $w_2' \subseteq w_2$. Die Formelmenge S heißt genau dann Kausalbasis von (w_1, w_2), wenn sie eine minimale Teilmenge ist von* $\mathrm{Diff}(w_1, w_2)$*, so daß gilt: $S \cup (w_1 \cap w_2) \vdash_C w_2'$ und $w_2' \cup B \vdash w_2$.*

Die Kausalbasis zweier möglicher Welten ist im allgemeinen nicht eindeutig. Als Tassenweltbeispiel stellen wir fest, daß die einzige Kausalbasis für (W_1, W_2) ist $\{\mathsf{Vorwärts}\}$. Das ergibt sich wie folgt: Es gilt

$$\{\mathsf{Vorwärts}\} \cup (W_1 \cap W_2) \vdash_{C_c} \{\mathsf{Regen}, \mathsf{Vorwärts}, \neg\mathsf{Rückwärts}, \neg\mathsf{Aufrecht}, \mathsf{Boden}\} =: W_2',$$

und weiterhin $W_2' \cup B_c \vdash W_2$; alle anderen Kausalbasis-Kandidaten enthalten $\mathsf{Vorwärts}$, sind also nicht minimal.

Bleibt nur noch zu definieren, wann eine mögliche Welt w_1 näher als eine andere w_2 an einer „Referenzwelt" w ist. Dafür gibt es intuitiv zwei, sich nicht ausschließende Möglichkeiten. w_1 ist näher an w als w_2, wenn seine Differenz zu w kleiner ist. Wenn die reine Differenz nicht kleiner ist, kann w_1 aber immer noch näher an w sein als w_2, wenn alle seine Kausalbasen kleiner sind. Präziser:

Definition 12 (Nähe, $\prec_w$) *Seien w, w_1 und w_2 mögliche Welten. w_1 ist genau dann näher an w als w_2, abgekürzt $w_1 \prec_w w_2$, wenn*

- $\mathrm{Diff}(w, w_1) \subset \mathrm{Diff}(w, w_2)$, *oder*
- *jede Kausalbasis von (w, w_1) ist Teilmenge einer Kausalbasis von (w, w_2), aber nicht andersherum.*

(Die beiden Bedingungen sind nicht äquivalent, und keine impliziert die jeweils andere. S. [Brewka und Hertzberg, 1993] für Beispiele.)

Als Tassenweltbeispiel wollen wir untersuchen, ob W_2 oder W_6 näher an W_1 ist. Zunächst: die einzige Kausalbasis von (W_1, W_6) ist $\{\neg\mathsf{Regen}, \mathsf{Vorwärts}\}$. Oben hatten wir gesehen, daß $\{\mathsf{Vorwärts}\}$ die einzige Kausalbasis von (W_1, W_2) ist; folglich gilt $W_2 \prec_{W_1} W_6$.

Schließlich werden Operatoren für MW definiert. Ich verwende verallgemeinerte STRIPS-Operatoren, wobei die Verallgemeinerung in zwei Ebenen läuft, wie weiter oben angedeutet: Die Anwendung eines Operators kann unterschiedliche, alternative Nachbedingungen erzeugen (man weiß vorher nicht, welche davon), und in unterschiedlichen Situationen kann ein Operator unterschiedliche Effekte erzeugen. Die Syntax eines Operators ist dann durch folgende Definition gegeben:

Definition 13 (Operator) *Sei $\mathcal{L}$ eine logische Sprache erster Stufe. Ein Operator ist ein Objekt, das nach folgendem Schema definiert ist:*

$$\begin{array}{lcll} [& Pre_1 & | & Post_{1,1}, \ldots, Post_{1,l(1)}; \\ & \vdots & & \\ & Pre_m & | & Post_{m,1}, \ldots, Post_{m,l(m)} \quad], \end{array}$$

Dabei sind die Vorbedingungen Pre_i und die Nachbedingungen $Post_{i,j}$ Formeln aus $\mathcal{L}$, die die folgende Bedingung erfüllen: $Pre_1 \vee \cdots \vee Pre_m$ ist logisch äquivalent zu TRUE*, und die Pre_i schließen sich wechselseitig aus.*

Zunächst zwei Beispiele von Operatoren außerhalb der Tassenwelt, welche die Ausdrucksmöglichkeiten klarmachen sollen.

- Eine Münze kann man immer (ohne weitere Vorbedingung) werfen; als Ergebnis kommt entweder Kopf oder Zahl heraus. Man beachte aber, daß das Ergebnis *nicht* als Kopf$\vee$Zahl angegeben werden darf, denn sinnvollerweise wird Kopf als äquivalent zu $\neg$Zahl definiert sein, so daß die einfache Disjunktion tautologisch wahr wäre. Der Münzwurf führt also in zwei unterschiedliche Nachfolgewelten, die nichts miteinander zu tun haben. Der entsprechende Operator sieht also so aus:

$$\textbf{MÜNZWURF} \;=\; [\ \textsc{true} \mid \textsf{Kopf}, \textsf{Zahl}\ ]$$

- Einen Tastenschalter fürs Raumlicht zu betätigen, hat unterschiedliche Effekte: War das Licht vorher An, ist es anschließend $\neg$An, und umgekehrt. Hier ist der entsprechende Operator:

$$\textbf{SCHALTEN} \;=\; \left[\ \begin{array}{c|c} \textsf{An} & \neg\textsf{An} \\ \neg\textsf{An} & \textsf{An} \end{array}\ \right]$$

In der Tassenwelt gebe es die folgenden Operatoren, deren formale Definitionen in Abbildung 10 angegeben sind:

ABSTELLE: Steht die Tasse auf dem Tisch, stelle sie Aufrecht auf den Boden. Der Operator kann (entgegen dieser Absicht) auch bewirken, daß sie Vorwärts auf dem Boden liegt (was mittels K_c bedeutet, daß $\neg$Tisch, $\neg$Rückwärts und $\neg$Aufrecht gelten). Die formale Definition in Bild 10 unterscheidet zusätzlich Anwendung bei Regen und $\neg$Regen, ohne daß der Unterschied sich auswirkt. Eine solche Auswirkung wird später (Abschnitt 6.2) kommen.

AUFRICHTE: Liegt die Tasse Rückwärts auf dem Boden, soll sie in Position Aufrecht verändert werden. Der Operator funktioniert sicher, wenn es nicht regnet; regnet es, kann auch sein, daß er scheitert, also nichts bewirkt. Liegt die Tasse nicht Rückwärts, bewirkt er keine Änderung.

DREHE: Liegt die Tasse auf dem Boden (Vorwärts oder Rückwärts), wird sie zufällig gedreht, so daß sie anschließend entweder Vorwärts oder Rückwärts liegt – aber natürlich unabhängig davon, wie sie vorher lag. In allen anderen Fällen bewirkt der Operator keine Änderung.

WARTE: Regnet es, kann nach Ende des Wartens es entweder nicht mehr regnen, oder es kann sich nichts geändert haben. Entsprechend für warten, wenn es nicht regnet.

Damit kann ich nun beschreiben, was eingangs dieses Abschnitts 5.1 als Ziel angegeben wurde: die Vorschrift, aus einer Situation s und einem Operator o die Nachfolgesituation zu berechnen. Die Idee dazu ist für MW sehr einfach: Man bildet die möglichen Welten in s ab in die nahesten möglichen Welten, in denen jeweils die $Post_{i,j}$ aus dem „Ast" von o gelten, der in der Ausgangswelt wahr ist. Ausführlich:

$$
\begin{array}{rcll}
\textbf{ABSTELLE} & = & [& \neg\mathsf{Regen} \wedge \mathsf{Tisch} \mid \mathsf{Aufrecht}, \mathsf{Vorwärts}; \\
 & & & \mathsf{Regen} \wedge \mathsf{Tisch} \mid \mathsf{Aufrecht}, \mathsf{Vorwärts}; \\
 & & & \neg\mathsf{Tisch} \mid \textsc{true} \quad] \\
\textbf{AUFRICHTE} & = & [& \neg\mathsf{Regen} \wedge \mathsf{Rückwärts} \mid \mathsf{Aufrecht}; \\
 & & & \mathsf{Regen} \wedge \mathsf{Rückwärts} \mid \mathsf{Aufrecht}, \textsc{true}; \\
 & & & \neg\mathsf{Rückwärts} \mid \textsc{true} \quad] \\
\textbf{DREHE} & = & [& \mathsf{Vorwärts} \vee \mathsf{Rückwärts} \mid \mathsf{Vorwärts}, \mathsf{Rückwärts}; \\
 & & & \neg(\mathsf{Vorwärts} \vee \mathsf{Rückwärts}) \mid \textsc{true} \quad] \\
\textbf{WARTE} & = & [& \mathsf{Regen} \mid \neg\mathsf{Regen}, \textsc{true}; \\
 & & & \neg\mathsf{Regen} \mid \mathsf{Regen}, \textsc{true} \quad]
\end{array}
$$

Abbildung 10: Die Operatoren für die Tassenwelt. Erklärungen im Text.

Definition 14 (Resultat eines Operators in s) *Sei K das durch C und B erzeugte Hintergrundwissen. Sei weiter s eine Formel und*

$$
\begin{array}{rcllll}
o & = & [& Pre_1 & \mid & Post_{1,1}, \ldots, Post_{1,l(1)}; \\
 & & & \vdots & & \\
 & & & Pre_m & \mid & Post_{m,1}, \ldots, Post_{m,l(m)} \quad],
\end{array}
$$

ein Operator. Das Ergebnis von o in s, abgekürzt $r(o,s)$, ist die Disjunktion aller möglicher Welten w', welche die folgende Bedingung erfüllen: Es gibt natürliche Zahlen $i \in \{1,\ldots,m\}$ und $j \in \{1,\ldots,l(i)\}$ und eine Welt $w \in \mathrm{Poss}_K(s)$, für die gilt, daß

1. *$w \models Pre_i$ und*
2. *w' ist $\prec_w$-minimal unter den möglichen Welten, die $Post_{i,j}$ erfüllen.*

Als Tassenweltbeispiel wollen wir $r(\textbf{ABSTELLE}, s_c)$ berechnen. (Zur Erinnerung: $s_c = \mathsf{Regen} \wedge (\mathsf{Tisch} \vee \mathsf{Vorwärts})$.) Wie wir gesehen haben, besteht $\mathrm{Poss}_{K_c}(s_c)$ aus den beiden möglichen Welten W_1, W_2. Wendet man **ABSTELLE** in W_2 an, bewirkt das keine Änderung, da hier $\neg$Tisch gilt, also der dritte Operator-„Zweig" angewendet wird. Wir halten also fest, daß W_2 im gesuchten Resultat enthalten ist.

In W_1 gilt Pre_2 von **ABSTELLE**. Für $Post_{2,2}$, also die Formel Vorwärts, haben wir alle notwendigen Berechnungen in den Beispielen vorher durchgeführt: Die einzige $\prec_{W_1}$-minimale Welt, die Vorwärts erfüllt, ist ebenfalls W_2. Das heißt, wendet man **ABSTELLE** in W_1 an, kann W_2 resultieren, wenn nämlich Vorwärts der „direkte" Effekt ist. Im Fall von $Post_{2,1}$, also der Formel Aufrecht, gibt es nur eine einzige mögliche Welt, die $\prec_{W_1}$-minimal ist und Aufrecht erfüllt, nämlich W_4. Zusammengefaßt gilt also:

$$r(\textbf{ABSTELLE}, s_c) = \{W_2, W_4\}.$$

Das schließt die Präsentation des Kalküls MW. Für einen Kalkül beispielsweise zur logischen Deduktion würde man erwarten, daß er mit der Semantik der entsprechenden

Logik in Beziehung gesetzt würde, also beispielsweise seine Korrektheit oder Vollständigkeit bewiesen würde. Das geht bei Aktionskalkülen nicht, da es derzeit keine formale Semantik von Handlungseffekten gibt. Immerhin läßt sich sagen, daß die von MW errechneten Ergebnisse der Intuition nicht grob zuwiderlaufen – das ist schon eine Menge. In der Terminologie von Abschnitt 2 formuliert fällt auf, daß MW, wiederum gemessen an der Intuition, das Vorhersageproblem löst.

5.2 Pläne für mögliche Welten

Nun ist die Untersuchung von Aktionskalkülen im allgemeinen und MW im besonderen hier ja nicht Selbstzweck, sondern die Kalküle sollen dazu dienen, den formalen Apparat für die Beschreibung von Plänen und von Planern zu liefern. Das ist, was im Rest dieses Textes geschieht.

Zunächst stellen wir dafür fest, daß wir die Definitionen von Plänen gegenüber Abschnitt 3 modifizieren müssen, denn eine geordnete Menge von Operatorinstanzen kann nicht beispielsweise die Effekt-Alternativen von MW-Operatoren darstellen. Ich beginne mit der Definition einer MW-Problembeschreibung, die sich von der STRIPS-Problembeschreibung (s. Def. 3) nur dadurch unterscheidet, daß sie das Hintergrundwissen berücksichtigt.

Definition 15 (MW-Problembeschreibung) *Sei $\mathcal{L}$ eine logische Sprache erster Ordnung. Eine MW-Problembeschreibung ist ein Quintupel $\langle s, O, f, C, B\rangle$; dabei sind*

$s \in \mathcal{L}$ eine Formel (Startsituation),

O eine Menge von MW-Operatorschemata,

$f \in \mathcal{L}$ eine Formel (Zielbeschreibung),

C eine Menge von Inferenzregeln über $\mathcal{L}$ (kausales Hintergrundwissen) und

$B \subseteq \mathcal{L}$ eine Menge von Formeln (logisches Hintergrundwissen).

Das Beispielproblem, das wir jetzt betrachten wollen, ist

$$\Psi_c = \langle s_c, \{\mathbf{ABSTELLE}, \mathbf{AUFRICHTE}, \mathbf{DREHE}, \mathbf{WARTE}\}, \mathsf{Aufrecht}, C_c, B_c\rangle.$$

Die Grundidee hinter der Darstellung von MW-Plänen ist, daß gar nicht erst mit beliebigen $\mathcal{L}$-Formeln gearbeitet wird, sondern ausschließlich mit möglichen Welten, wobei ein Operator von einer möglichen Welt gemäß der $r(\cdot,\cdot)$-Relation in mehrere mögliche Welten führen kann. Da es nur endlich viele mögliche Welten gibt (die möglicherweise nicht einmal alle durch Operatoranwendungen aus den möglichen Welten der Startsituation erreichbar sind), kann man sich einen Plan als einen nichtdeterministischen endlichen Automat vorstellen, dessen Zustände mögliche Welten sind und bei dem Operatorinstanzen Zustandsübergänge bewirken. Die möglichen Welten der Startsituation werden durch einen Pseudo-Operator $\mathcal{I}$ erzeugt, der als zur Operatormenge jeder MW-Problembeschreibung dazugehörig definiert sei. Bezeichne ich Operatoren in den Plänen als T-Knoten (T für *task*) und mögliche Welten als W-Knoten, kommt die folgende Definition heraus; man überzeuge sich, daß ein MW-Plan eine Spezialisierung der Begriffsbestimmung 1 von Plänen im allgemeinen ist.

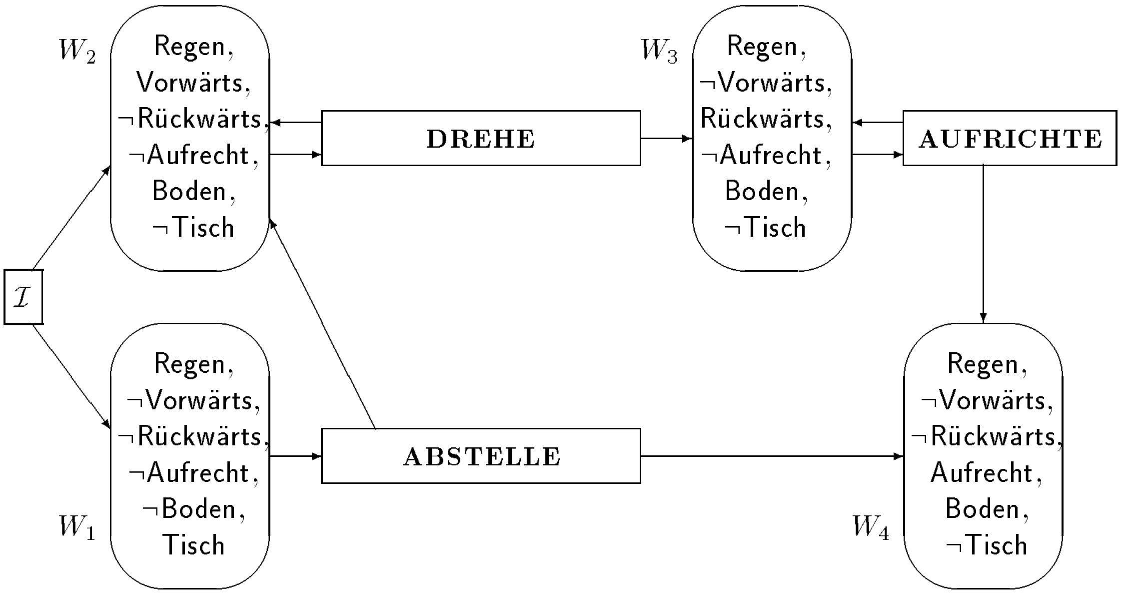


Abbildung 11: Der Plan Π_c für Ψ_c. T-Knoten sind durch eckige Kästchen dargestellt, W-Knoten durch Kästchen mit gerundeten Ecken.

Definition 16 (MW-Plan) *Sei $\Psi = (s, O, f, C, B)$ eine MW-Problembeschreibung, und sei $\mathcal{I} = [\text{TRUE} \mid w_1, \ldots, w_n]$ ein Operator, wobei $\{w_1, \ldots, w_n\} = \text{Poss}_K(s)$. Ein Plan Π für Ψ ist ein bipartiter gerichteter Graph aus etikettierten T- und W-Knoten mit einer eindeutigen Wurzel, wobei gilt:*

1. *Die Wurzel ist ein T-Knoten namens START; sie ist etikettiert mit $\mathcal{I}$. Alle anderen T-Knoten sind etikettiert mit Instanzen von Operatoren aus O.*
2. *Jeder W-Knoten ist etikettiert mit einem Element aus $Poss(K)$. Keine zwei W-Knoten sind mit derselben Welt etikettiert.*
3. *Alle Blätter sind W-Knoten.*
4. *T-Knoten haben ausschließlich W-Knoten als Nachfolger.*
5. *Jeder W-Knoten, der kein Blatt ist, hat genau einen T-Knoten als Nachfolger. Jeder T-Knoten außer START hat genau einen W-Knoten als Vorgänger.*
6. *Von jedem Knoten führt mindestens ein Weg zu einem Blatt.*

Abbildung 11 zeigt einen Plan für die Problembeschreibung Ψ_c. Der Plan ist folgendermaßen zu interpretieren: Ist die Welt W_1 diejenige, die anfangs vorliegt (was man erst zur Ausführungszeit feststellen kann), dann wende den Operator **ABSTELLE** an. Wie weiter oben erklärt, können sich dann als Resultatwelten W_2 oder W_4 ergeben; entsprechend sind die entsprechenden Welten Nachfolger dieses T-Knotens. Π_c hat ein einziges Blatt, nämlich W_4. Der Teilplan bestehend nur aus dem T-Knoten $\mathcal{I}$ und W_1, W_2 hat zwei Blätter, nämlich W_1, W_2.

Π_c ist insofern ein Ausnahmeplan, als er tatsächlich alle Nachfolgewelten aller Operatoren in seinen T-Knoten korrekt darstellt und an seinem Ende, also im Blatt W_4, sogar die Zielbedingung der zugrundeliegenden Problembeschreibung gilt. Π_c ist in diesem Sinne eine Lösung für Ψ_c. Definition 16 fordert das nicht, aber an einen „fertigen" MW-Plan müßte man diese Forderung stellen. Im klassischen Planen war das Kriterium für „sinnvolle" Pläne Wohlgeformtheit und Korrektheit; hier will ich das wie gesagt relativ zum Kalkül MW definieren. Die entsprechenden Begriffe sind *Korrektheit* und *Vollständigkeit* eines Plans relativ zu MW. Ich beginne mit Korrektheit.

Korrektheit eines MW-Plans bedeutet intuitiv: Sagt ein Plan, daß etwas der Fall ist oder sein kann, so ist oder kann es tatsächlich der Fall sein. Insbesondere bedeutet das:[25]

1. Wenn die Planausführung einen Pfad im Plan entlangläuft, dann sagt der Plan richtig vorher, welche Nachfolgewelten entstehen können;
2. jede Handlung, die einem Operator im Plan entspricht, ist tatsächlich ausführbar; und
3. wenn der Plan sagt, daß nun keine Handlung mehr ausgeführt zu werden braucht, dann ist ein Zielzustand erreicht.

Ich will hier allerdings nur *partielle* Korrektheit fordern in dem Sinn, daß die Zielformel gilt, wenn der Plan angibt, in einer möglichen Welt keinen weiteren Operator anzuwenden – im allgemeinen kann man nicht garantieren, daß jede Ausführung eines jeden korrekten Plans auch tatsächlich in einem Zielzustand endet. Das liegt daran, daß Pläne Schleifen wie zum Beispiel die mit dem **DREHE**-Operator in Π_c enthalten können, aus denen syntaktisch ein Weg herausführt; doch keiner kann garantieren, daß die Planausführung *tatsächlich* aus dieser Schleife herauskommt. Wiederum ist das ein Problem der Bereichsmodellierung, das außerhalb der Möglichkeiten der formalen Betrachtung liegt.

Nach diesen Vorüberlegungen wird Plankorrektheit nun zunächst für einzelne Knotentypen definiert, und dann für Pläne insgesamt.

Definition 17 (Korrektheit von MW-Plänen) *Sei* Π *ein* MW*-Plan für die Problembeschreibung* $\Psi = \langle s, O, f, C, B \rangle$.

1. *(Korrektheit der Wurzel.) Die Planwurzel* START *ist korrekt bzgl.* Ψ*, gdw. für jeden Nachfolge-W-Knoten von* START*, der mit der möglichen Welt* w *etikettiert ist, gilt:* $w \in \mathrm{Poss}_K(s)$
2. *(Korrektheit der anderen T-Knoten.) Sei* $\tau \neq$ START *ein T-Knoten, etikettiert mit* o*, und sei* ω *sein Vorgänger, der mit* w *etikettiert sei.* τ *ist korrekt bzgl.* Ψ*, gdw. für jeden Nachfolger* ω' *von* τ*, der mit* w' *etikettiert ist, gilt, daß* $w' \in r_K(o, w)$.
3. *(Korrektheit von Blattknoten.) Ein Blatt in* Π*, das mit* w *etikettiert ist, ist korrekt bzgl.* Ψ*, gdw.* $w \cup K \vdash f$.
4. *(Plankorrektheit.)* Π *ist korrekt bzgl.* Ψ*, gdw.*
   - *alle T-Knoten von* Π *sind korrekt bzgl.* Ψ *und*

[25] Alles das kann strenggenommen natürlich nur garantiert werden, wenn die Problembeschreibung korrekt ist bezüglich des Anwendungsbereichs. Das ist der Sache nach formal natürlich nicht verifizierbar und liegt in der Verantwortung der Bereichsmodellierer. Ich klammere es daher hier aus.

- *alle Blattknoten von* Π *sind korrekt bzgl.* Ψ.

Vollständigkeit von MW-Plänen ist analog zu Korrektheit definiert. Die folgende Intuition unterliegt diesem Begriff: Wenn ein Effekt bei der Ausführung einer Handlung in einem Zustand herauskommen kann, dann ist dieser Effekt im Plan auch bereits berücksichtigt.[26] Das bedeutet insbesondere, daß der Plan alle möglichen Welten der Startsituation enthalten muß, sowie alle möglichen Welten, die aus der Anwendung eines im Plan enthaltenen Operators resultieren können. Diese Forderung bedeutet natürlich *nicht,* daß der Plan für jede mögliche Welt im Bereich einen Operator enthalten muß. Da ein Plan eben ein Plan ist, muß er nur sagen, was in den Situationen zu tun ist, die bei seiner Ausführung vorhersehbar auftreten werden.

Definition 18 (Vollständigkeit von MW-Plänen) *Für die Problembeschreibung* $\Psi = \langle s, O, f, C, B\rangle$ *sei* Π *ein* MW-*Plan.*

1. *(Vollständigkeit der Wurzel.) Die Planwurzel* START *ist vollständig bzgl.* Ψ, *gdw. für alle* $w \in \mathrm{Poss}_K(s)$ *es einen Nachfolger von* START *gibt, der mit* w *etikettiert ist.*
2. *(Vollständigkeit der anderen T-Knoten.) Sei* $\tau \neq$ START *ein T-Knoten, etikettiert mit* o, *und sei* ω *sein Vorgänger, der mit* w *etikettiert sei.* τ *ist vollständig bzgl.* Ψ, *gdw. es für alle* $w' \in r_K(o, w)$ *einen Nachfolger von* τ *in* Π *gibt, der mit* w' *etikettiert ist.*
3. *(Planvollständigkeit.)* Π *ist vollständig bzgl.* Ψ, *gdw. alle T-Knoten vollständig sind bzgl.* Ψ.

In der Tassenwelt ist Π_c ein Beispiel für einen Plan, der korrekt und vollständig ist bzgl. Ψ_c.

Korrektheit und Vollständigkeit von Plänen sind wichtige Begriffe fürs Planen. Zum Beispiel können sie im Prinzip als Abbruchkriterien für eine Planugnsprozedur verwendet werden: ein Plan ist „fertig“, wenn er korrekt und vollständig ist.[27] Doch für den Bau praktischer Planer sind Korrektheit und Vollständigkeit in der angegebenen, strikten Form unbrauchbar. Da sie über MW an etlichen Stellen erfordern zu prüfen, ob eine im Prinzip beliebige prädikatenlogische Formel aus einer Formelmenge folgt, sind sie nicht einmal entscheidbar, geschweige denn effizient berechenbar. Weiterhin ist es zumindest diskutierbar, ob man die hier definierten „strikten“ Versionen von Korrektheit und Vollständigkeit überhaupt haben will: Wenn MW sehr liberal ist, was Unsicherheit der Problembereichsbeschreibung angeht (die Startsituation muß nicht genau beschrieben sein, die Nachbedingungen müssen nicht eindeutig sein), dann scheint die Forderung etwas abseitig, daß diese unsichere Information aber gnadenlos korrekt und vollständig ausgeschöpft zu werden hat – ohne Rücksicht auf Rechenzeit.

Thema des abschließenden Abschnitts ist folglich die Frage, wie praktische Planer auf „sanftere“ Versionen von Plan-Korrektheit und -Vollständigkeit aufgebaut werden können, die immer noch den klaren Bezug zum zugrundeliegenden Kalkül – im Beispiel also MW – haben, aber dennoch weniger rigide sind.

[26] Wieder setze ich natürlich voraus, daß die verwendete Problembeschreibung intuitiv korrekt ist bzgl. des Bereichs.

[27] In der Prozedur NLP in Abbildung 3 sind an entsprechender Stelle Wohlgeformtheit und Konfliktfreiheit verwendet worden. Es wäre zu zeigen, daß ein wohlgeformter und konfliktfreier klassischer Plan immer korrekt und vollständig ist, eine entsprechende Definition davon für klassische Pläne vorausgesetzt.

6 „Begrenzt-korrekte“ Planer

Grundsätzlich sind viele Möglichkeiten denkbar, „sanftere“ Versionen von Vollständigkeit und Korrektheit zu definieren. *Eine* solche Möglichkeit könnte zum Beispiel sein, in Analogie zu dem im Maschinellen Lernen bekannten PAC-Lernen[28] „PAC-Planen“ zu betreiben: Die Idee wäre, mit polynomiell zeitbeschränkten Algorithmen vollständige und korrekte Pläne zu *approximieren,* wobei die Laufzeit von der Genauigkeit der Approximation abhängt.

Die Idee, die ich hier verwende, ist ähnlich, aber anders: Ich werde eine *anytime*-Planungsprozedur entwerfen, wie in Abschnitt 4 skizziert, und zwar genauer eine *unterbrechbare* solche Prozedur. Der Plan, den diese Prozedur liefert, soll also umso korrekter und vollständiger werden, je länger man ihr zu laufen erlaubt – bis hin zu einem Optimalwert, jenseits dessen sich nichts mehr verbessern kann. Allerdings machen ich eine Einschränkung: approximiert wird lediglich Korrektheit, nicht Vollständigkeit. Das ist nicht notwendig so, aber in dem Beispiel, das auf die MW-Pläne des vorigen Abschnitts aufbaut, liegt es nahe.

Die Grundsatzidee für einen unterbrechbaren Planer für MW-Pläne ist sehr einfach. Abbildung 12 zeigt das Schema einer entsprechenden Prozedur. Eingabe sind eine MW-Problembeschreibung und ein MW-Plan, wobei dieser Plan beim initialen Prozeduraufruf aus dem T-Knoten START mit den W-Knoten-Nachfolgern besteht, die den möglichen Welten im Startzustand entsprechen. Die Prozedur läuft im Prinzip endlos – der Tatsache entsprechend, daß es unendlich viele unterschiedliche Operatorinstanzen für Schritt 1. geben kann, die alle durchprobiert werden können, wenn die Rechenzeit unbegrenzt ist. Der derzeit beste Plan, also gleichsam das derzeitige Ergebnis der Prozedur, ist in der globalen Variablen *best-plan* abgelegt. Implizit ist angenommen, daß die Prozedur **fail** liefert, wenn in den Schritten 0. oder 1. keine Wahlmöglichkeit (mehr) besteht; in diesem Fall wird zum vorigen Auswahlpunkt zurückgesprungen und dort die nächste Wahlmöglichkeit bearbeitet.

Wie beim klassischen Planen mit der Prozedur NLP ist offenbar auch bei MWPU die Auswahlstrategie an den Wahlpunkten für das Laufverhalten der Prozedur entscheidend. Doch es gibt einen weiteren Parameter, der das Verhalten beeinflußt: die Funktion QUALITY. Soll sichergestellt sein, daß durch MWPU mit steigender Laufzeit der *best-plan* immer „korrekter“ wird, muß die Funktion QUALITY das widerspiegeln. Das bedeutet aber: In QUALITY muß die Definition eingehen, was es heißt, daß ein MW-Plan „korrekter“ ist als ein anderer.

Während sich die Struktur der Prozedur MWPU nicht von einer entsprechenden unterbrechbaren Prozedur zum Erzeugen klassischer Pläne unterscheidet, liegt in QUALITY das MW-Spezifische. Daraus folgt: Ich hätte eine begrenzt-korrekte Planungsprozedur auch für die Erzeugung klassischer Pläne angeben und im folgenden untersuchen können, oder auch für Pläne unter allen anderen Voraussetzungen an den Anwendungsbereich. Daß ich diese Untersuchung an den etwas komplizierteren MW-Plänen vorführe, liegt daran, daß Begrenzt-Korrektheit dazu besser paßt als zu klassischen Plänen: Es macht viel Sinn, nicht auf strikter Plankorrektheit zu bestehen, wenn die Bereichsdarstellung als unsicher und unvollständig vorausgesetzt ist wie in MW und in MW-Plänen; für klassische Pläne mit ih-

[28] *probably approximately correct learning,* s. [Hoffmann, 1991] für eine Kurzübersicht und weitere Literatur.

Prozedur MWPU(Π, Ψ)

Eingabe: Π: MW-Plan
$\Psi = \langle s, O, f, C, B \rangle$: MW-Problembeschreibung

Ausgabe: –

global: *best-plan*: bisher bester gefundener Plan.
initial: Initialplan

---

0. wähle einen Blattknoten ω, der mit w etikettiert ist, wobei w keine Ziel-Welt ist;
1. wähle eine Operatorinstanz o;
2. füge einen mit o etikettierten T-Knoten in Π als Nachfolger von ω ein;
   markiere jeden W-Knoten ω', der mit einer möglichen Welt aus $r(o, w)$ etikettiert ist, als Nachfolger von o (füge ggf. entsprechende neue W-Knoten ein);
   der neue Plan heiße Π';
3. **if** QUALITY(Π') > QUALITY(*best-plan*) **then atomic**[*best-plan* := Π'];
4. **call** MWPU(Π', Ψ)

Abbildung 12: Eine unterbrechbare Prozedur zum Erzeugen von MW-Plänen; Erklärung im Text.

rer voraussetzungsgemäß vollständigen Information macht es möglicherweise weniger Sinn. Aber ich will nicht verschweigen, daß ich der Meinung bin, es sei durchaus der Mühe wert, über begrenzt-korrekte Versionen des klassischen Planens nachzudenken – möglicherweise verlieren die Ergebnisse zur theoretischen Komplexität beispielsweise des aussagelogischen Planens (s. Abschnitt 3) etwas von ihrem Schrecken.

Im folgenden werden zunächst allgemeine Konzepte und Definitionen vorgestellt, die für das Verständnis von Begrenzt-Korrektheit erforderlich sind. Anschließend werde ich als Beispiel einen begrenzt-korrekten Planer zur Erzeugung von MW-Plänen entwickeln.

6.1 Definition der Begrenzt-Korrektheit

Nach Definition ist das Ergebnis eines begrenzt-korrekten Planers im allgemeinen und eines unterbrechbaren im besonderen laufzeitabhängig. Bislang hatte ich Planer als Einheiten angesehen, die eine Problembeschreibung bekommen und einen Plan zurückgeben; für unterbrechbare begrenzt-korrekte Planer muß diese Funktionalität also laufzeitabhängig gemacht werden. Das geschieht in der folgenden Definition, in deren Sinn ich den Begriff Planer für den Rest dieses Abschnitts verwenden werde.

Definition 19 (Planer) *Sei Ψ eine* MW-*Problembeschreibung. Ein Planer ist ein Paar $\langle \mathcal{P}, A_{\mathcal{P}} \rangle$ aus folgenden Funktionen:*

$\mathcal{P}$ *ist eine Funktion vom Typ* `Problembeschreibung` $\rightarrow I\!N \rightarrow$ `Plan`*; $\mathcal{P}$ heißt Plan-Erzeugungs-Funktion.*

$A_\mathcal{P}$ *ist eine totale Funktion des Typs* `Problembeschreibung` $\rightarrow 2^{\mathbb{N}}$, *so daß* $\mathcal{P}(\Psi)$ *eine linkstotale Funktion auf* $A_\mathcal{P}(\Psi) \subseteq \mathbb{N}$ *für jedes Problem* Ψ *ist;* $A_\mathcal{P}$ *heißt Verfügbarkeitsfunktion.*

Dieser Definition liegt folgende Vorstellung zugrunde. Die verbrauchte Laufzeit des Planers wird in natürlichen Zahlen einer frei zu wählenden Einheit („Ticks“) gemessen. Im allgemeinen muß ein zeitabhängiger Planer nicht nach jeder beliebigen Zahl von Zeiteinheiten einen Plan abliefern können; daher gibt es die Verfügbarkeitsfunktion, die problembeschreibungsabhängig angibt, nach welchen Zahlen von Zeiteinheiten der Planer ein Ergebnis verfügbar haben wird. (Diese Funktion ist ein theoretisches Konstrukt – sie muß nicht effektiv angebbar sein.) Die „eigentliche“ Planfunktion $\mathcal{P}$ wird mit einer Problembeschreibung und einer Zahl von Zeiteinheiten aufgerufen, für die $A_\mathcal{P}$ garantiert, daß $\mathcal{P}$ einen Plan dafür findet; diesen Plan liefert sie als Ergebnis.

Ein unterbrechbarer *anytime*-Planer ist, wie weiter oben intuitiv gesagt, ein Planer, der nach *jeder* Zahl von Zeiteinheiten einen Plan zurückgibt (der nicht notwendig korrekt oder vollständig sein muß). Für einen solchen Planer muß also gelten, daß $A_\mathcal{P} = \mathbb{N}$.

Ein *strikt* korrekter Planer ist dann ein Planer, der nur korrekte und vollständige Pläne zurückgibt, wenn er welche zurückgibt. Strikt korrekte Planer sind in diesem Abschnitt nicht das Thema, aber zum Vergleich mit der dann zu entwickelnden Definition von Begrenzt-Korrektheit definiere ich dennoch strikte Korrektheit:

Definition 20 ((Strikte) Planerkorrektheit) *Ein Planer* $\langle \mathcal{P}, A_\mathcal{P} \rangle$ *ist korrekt, gdw. der Plan* $\mathcal{P}(\Psi)(t)$ *korrekt und vollständig bzgl.* Ψ *ist für alle lösbaren Probleme* Ψ *und für alle* $t \in A_\mathcal{P}(\Psi)$.

Ich definiere nun eine Planbewertungsfunktion, die sowohl dazu dient, strikte von begrenzter Planerkorrektheit zu unterscheiden, als auch eingehen kann in die noch ausstehende, vom konkreten Planformat abhängige Definition der QUALITY-Funktion. Ich fordere eine Funktion, die einen (wiederum für unterschiedliche Plan-Formate unterschiedlich zu definierenden) intuitiven „Grad“ an Plankorrektheit abbildet auf ein $[0, 1]$-Interval, wobei genau solche Pläne mit 1 bewertet werden, die korrekt und vollständig sind.

Definition 21 (Planbewertungsfunktion) *Sei* Ψ *eine Problembeschreibung. Eine Planbewertungsfunktion für* Ψ *ist eine totale Funktion* β_Ψ, *die Pläne für* Ψ *auf* $[0, 1]$ *abbildet, so daß für jeden Plan* Π *gilt:*

$$\beta_\Psi(\Pi) = 1 \textit{ gdw. } \Pi \textit{ ist korrekt und vollständig bzgl. } \Psi$$

Ein strikt korrekter Planer würde offenbar nur solche Pläne abliefern, die mit 1 bewertet sind. Genau diese strenge Forderung soll für Begrenzt-Korrektheit nicht mehr gelten: „eine Zeitlang“ möchte man auch Pläne akzeptieren, die nicht strikt korrekt sind; allerdings müssen zwei Dinge sichergestellt sein:

1. irgendwann muß ein korrekter Plan geliefert werden, wenn es einen gibt, und
2. die Bewertung der gelieferten Pläne soll mit Laufzeit des Planers monoton wachsen – ein einmal erreichter „Grad an Korrektheit“ soll also nicht wieder unterschritten werden.

Genau diese beiden Punkte drückt die folgende Definition aus. Dazu noch eine Erläuterung: Es ist darin von einer „Familie von Planbewertungsfunktionen" die Rede. Das ist erforderlich, weil Planbewertungsfunktionen nur für einzelne Problembeschreibungen definiert sind, Begrenzt-Korrektheit aber für Planer für komplette Problem*bereiche* definiert werden sollen, in denen es also viele einzelne Problembeschreibungen gibt. Die entsprechenden Verallgemeinerung der Planbewertungsfunktion ist rein technischer Natur.

Definition 22 (Begrenzt-Korrektheit) *Sei β eine Familie von Planbewertungsfunktionen. Ein Planer $\Phi = \langle \mathcal{P}, A_{\mathcal{P}} \rangle$ ist begrenzt korrekt bzgl. β, gdw. für alle Probleme Ψ gilt:*

1. *wenn es einen Plan gibt, der korrekt und vollständig bzgl. Ψ ist, dann wird Φ in endlicher Zeit einen korrekten und vollständigen Plan erzeugen, d.h. dann existiert $t \in A_{\mathcal{P}}(\Psi)$, so daß $\beta_\Psi(\mathcal{P}(\Psi)(t)) = 1$, und*

2. *die Bewertungen der Pläne, die Φ erzeugt, wachsen monoton mit der Planungszeit, d.h. für alle $t_1, t_2 \in A_{\mathcal{P}}(\Psi)$: wenn $t_2 \geq t_1$ dann $\beta_\Psi(\mathcal{P}(\Psi)(t_2)) \geq \beta_\Psi(\mathcal{P}(\Psi)(t_1))$*

Das Konzept Begrenzt-Korrektheit ist offenbar unabhängig vom betrachteten Planformat: beispielsweise könnte man begrenzt-korrekte Versionen klassischer Planer definieren. Allerdings ist dieses Konzept in Kombination mit unterschiedlichen Merkmalen des Anwendungsbereichs unterschiedlich sinnvoll. Besonders sinnvoll scheint es da zu sein, wo die Information über den Anwendungsbereich als unvollständig oder unsicher vorausgesetzt ist, also beispielsweise für MW-Pläne.

6.2 Ein begrenzt-korrekter MW-Planer

Begrenzt-Korrektheit eines Planers zur Erzeugung von MW-Plänen erreiche ich nun auf die folgende Weise. Ich definiere für einen MW-Plan Π den

Korrektheitsgrad: der Anteil an Nachfolgewelten eines Operatorknotens in Π, die nicht Operatoreffekten laut dem Kalkül MW entsprechen;

Vollständigkeitsgrad: der Anteil an möglichen Effekten eines Operators laut MW, für die Π eine entsprechende Folgeaktion enthält

Damit käme man aus. Weil es aber so schön dazupaßt, will ich gleich noch eine von Begrenzt-Korrektheit eigentlich unabhängige Idee ansatzweise mitverarbeiten, die im Abschnitt 4 angesprochen wurde: entscheidungstheoretisches Planen. Ich werde die Möglichkeit schaffen, Information über a-priori Wahrscheinlichkeiten der Gültigkeit von Formeln und des Eintretens von Operator-Nachbedingungen anzugeben, die sich auf die möglichen Welten im MW-Plan übertragen. Im Effekt können wir dann nicht nur sagen, wieviele Welten beispielsweise noch nicht im Plan berücksichtigt sind, sondern auch, wie wahrscheinlich es überhaupt ist, daß sie eintreten könnten; diese Information wiederum kann in die Festlegung des Grades an Korrektheit und Vollständigkeit eines MW-Plans eingehen. (Weiterhin kann eine Planungsprozedur sie verwenden, um die „offene Stelle" als nächste zu bearbeiten, die bei Ausführung maximal wahrscheinlich erreicht wird – was hier aber nicht interessiert.)

$W \in \mathrm{Poss}_{K_c}$		$p(W)$
W_1	{ Regen, ¬Vorwärts, ¬Rückwärts, ¬Aufrecht }	0.08
W_2	{ Regen, Vorwärts, ¬Rückwärts, ¬Aufrecht }	0.14
W_3	{ Regen, ¬Vorwärts, Rückwärts, ¬Aufrecht }	0.14
W_4	{ Regen, ¬Vorwärts, ¬Rückwärts, Aufrecht }	0.04
W_5	{ ¬Regen, ¬Vorwärts, ¬Rückwärts, ¬Aufrecht }	0.12
W_6	{ ¬Regen, Vorwärts, ¬Rückwärts, ¬Aufrecht }	0.21
W_7	{ ¬Regen, ¬Vorwärts, Rückwärts, ¬Aufrecht }	0.21
W_8	{ ¬Regen, ¬Vorwärts, ¬Rückwärts, Aufrecht }	0.06

Abbildung 13: A-priori Wahrscheinlichkeitsverteilung für die Tassenwelt unter P_c.

Die volle Definition der dahinterstehenden Konzepte ist technisch ein wenig aufwendig und sprengt den Rahmen dieses Abschnitts, weshalb ich sie hier nicht im Detail angeben kann. Für genauere Information sei verwiesen auf [Hertzberg und Thiébaux, 1993].

Die erste Form von vorzugebender Wahrscheinlichkeitsinformation ist die über eine a-priori-Verteilung von Formeln. Weiß man beispielsweise, daß es ohne weitere Information in 40% aller Fälle regnet, kann das so angegeben werden. Diese Information wird dann umgesetzt in eine Wahrscheinlichkeitsverteilung auf den möglichen Welten, die so geartet ist, daß die a-priori-Wahrscheinlichkeiten für Formeln erfüllt sind (wenn sie erfüllbar waren) – es geht also *nicht* darum, unendlich viele oder *fuzzy* Wahrheitswerte einzuführen. Wird keine solche a-priori-Information gegeben, werden die möglichen Welten als gleichverteilt angenommen. Die Wahrscheinlichkeitsinformation wird in Form einer Verteilung P, dem *probabilistischen Hintergrundwissen,* angegeben. Ein Tassenwelt-Beispiel ist

$$P_c = \{\ p(\mathsf{Regen}) = 0.4,\ p(\mathsf{Vorwärts} \vee \mathsf{Rückwärts}) = 0.7,\ p(\mathsf{Tisch}) = 0.2\ \}$$

Damit ergibt sich formal:

Definition 23 (A-priori-Wahrscheinlichkeitsverteilung) *Seien K und P das logische und das probabilistische Hintergrundwissen. Die a-priori-Wahrscheinlichkeitsverteilung p über dem Raum der möglichen Welten ist diejenige Verteilung, für die gilt*

$$p(true) = 1 = \sum_{w \in \mathrm{Poss}(K)} p(w)$$

$$\forall p(s) \in P. \left[p(s) = \sum_{w \in \mathrm{Poss}_K(s)} p(w) \right]$$

und für die die Entropie maximal ist, die definiert ist als $-\sum_{w \in \mathrm{Poss}(K)} p(w) \log p(w)$.

Die a-priori-Verteilung auf den möglichen Welten der Tassenwelt unter P_c ist in Abbildung 13 angegeben. (Die W_i sind dieselben wie in Abschnitt 5.)

In naheliegender Weise wird dann die abhängige Wahrscheinlichkeit definiert, daß eine mögliche Welt w vorliegt, sofern eine Formel s wahr ist:

$$
\begin{array}{rlr|ll}
\textbf{ABSTELLE} = & [& \neg\mathsf{Regen} \wedge \mathsf{Tisch} & (\mathsf{Aufrecht}, 0.8),\ (\mathsf{Vorwärts}, 0.2); & \\
& & \mathsf{Regen} \wedge \mathsf{Tisch} & (\mathsf{Aufrecht}, 0.6),\ (\mathsf{Vorwärts}, 0.4); & \\
& & \neg\mathsf{Tisch} & (\textsc{true}, 1) &] \\
\textbf{AUFRICHTE} = & [& \neg\mathsf{Regen} \wedge \mathsf{Rückwärts} & (\mathsf{Aufrecht}, 1); & \\
& & \mathsf{Regen} \wedge \mathsf{Rückwärts} & (\mathsf{Aufrecht}, 0.8),\ (\textsc{true}, 0.2); & \\
& & \neg\mathsf{Rückwärts} & (\textsc{true}, 1) &] \\
\textbf{DREHE} = & [& \mathsf{Vorwärts} \vee \mathsf{Rückwärts} & (\mathsf{Vorwärts}, 0.5),\ (\mathsf{Rückwärts}, 0.5); & \\
& & \neg(\mathsf{Vorwärts} \vee \mathsf{Rückwärts}) & (\textsc{true}, 1) &] \\
\textbf{WARTE} = & [& \mathsf{Regen} & (\neg\mathsf{Regen}, 0.1),\ (\textsc{true}, 0.9); & \\
& & \neg\mathsf{Regen} & (\mathsf{Regen}, 0.1),\ (\textsc{true}, 0.9) &]
\end{array}
$$

Abbildung 14: Die Operatoren der Tassenwelt mit assoziierten Wahrscheinlichkeiten für Nachbedingungen.

Definition 24 (Revidierte Wahrscheinlichkeitsverteilung) *Sei p die a-priori-Wahrscheinlichkeitsverteilung über dem Raum der möglichen Welten bei K und P, und sei s eine Formel. Die Revision von p unter s, notiert als p_s, ist gegeben durch*

$$
p_s(w) = \begin{cases} \frac{p(w)}{\sum_{w' \in \mathrm{Poss}_K(s)} p(w')} & \textit{falls } w \in \mathrm{Poss}_K(s), \\ 0 & \textit{sonst.} \end{cases}
$$

Als Tassenwelt-Beispiel betrachten wir

$$
\begin{array}{lclclcl}
p_{s_c}(W_1) & = & \frac{p(W_1)}{p(W_1)+p(W_2)} & = & \frac{0.08}{0.22} & \simeq & 0.36 \\
p_{s_c}(W_2) & = & \frac{p(W_2)}{p(W_1)+p(W_2)} & = & \frac{0.14}{0.22} & \simeq & 0.64
\end{array}
$$

Als zweite Information muß vorgegeben werden, welche Nachbedingungen eines Zweiges eines Operators mit welcher relativen Wahrscheinlichkeit auftreten. Dafür muß das Format zur Definition von Operatoren entsprechend erweitert werden. Das tue ich in der naheliegenden Weise und gebe statt einer Definition einfach die neuen Versionen der Operatoren der Tassenwelt in Abbildung 14 an. Wiederum kann auch keine Information angegeben sein; in diesem Fall wird Gleichverteilung der Effekte innerhalb eines Astes angenommen.

Nun bleibt festzulegen, wie die Wahrscheinlichkeitsinformation verrechnet wird, wenn sich mögliche Welten als Operatoreffekte ergeben. Wendet man einen Operator o in einer möglichen Welt w an, ist die Wahrscheinlichkeit, daß w' herauskommt, einfach die Summe der bedingten Wahrscheinlichkeiten, daß w' Resultat eines der Zweige aus dem in w angewendeten Operator-Ast von o ist, multipliziert mit der Wahrscheinlichkeit für den entsprechenden Zweig. Allgemeiner: Wendet man o in einer Situation an, die nur durch eine Formel s beschrieben ist, muß man alle möglichen Welten von s berücksichtigen und mit deren Wahrscheinlichkeiten gewichtet summieren:

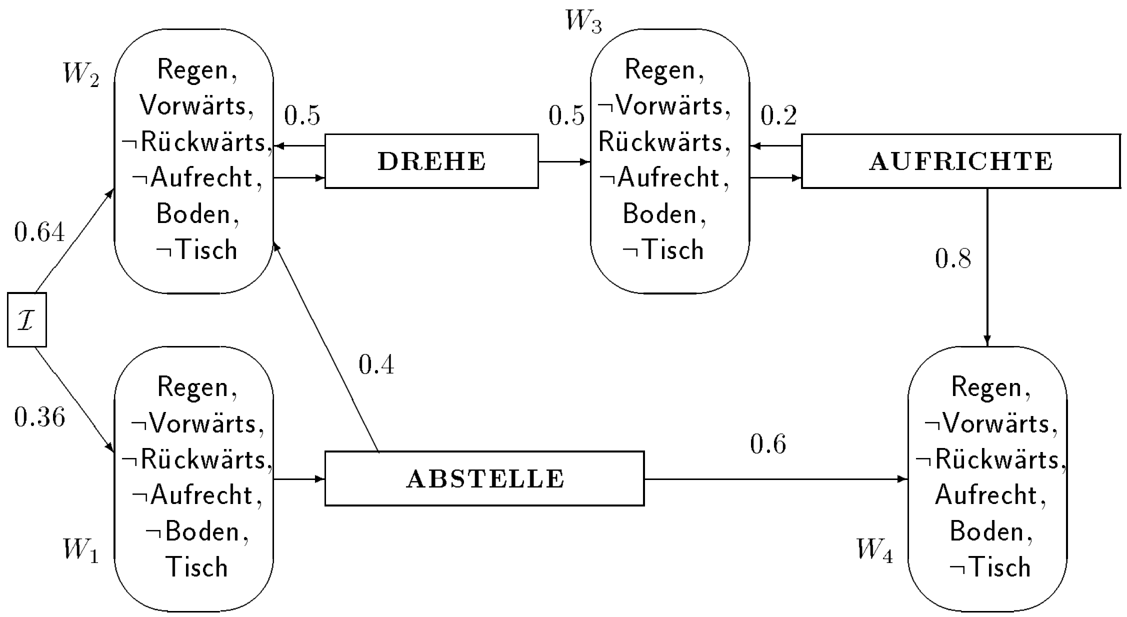


Abbildung 15: Der Plan Π_c mit zusätzlicher Wahrscheinlichkeitsinformation.

Definition 25 (Resultierende Wahrscheinlichkeitsverteilung) *Sei p die a-priori-Verteilung über dem Raum der möglichen Welten bei K und P, sei $s \in \mathcal{L}$ eine Formel und o ein Operator. Die Wahrscheinlichkeitsverteilung, die aus Anwendung von o in s resultiert, notiert als $p_{(o,s)}$, ist gegeben durch*

$$p_{(o,s)}(w') = \sum_{w \in \mathrm{Poss}_K(s)} \left(p_s(w) \sum_{j=1}^{l(i)} p(w' \mid \prec_w\text{-minimal}(Post_{i,j}))\ p_{i,j} \right),$$

wobei $K \cup w \vdash Pre_i$, und $p(w' \mid \prec_w\text{-minimal}(Post_{i,j}))$ ist die Wahrscheinlichkeit für w' unter der Bedingung, daß $Post_{i,j}$ der Effekt der Anwendung von o in w ist.

In der Tassenwelt gilt beispielsweise:

$$\begin{array}{rcl} p_{(table2up,s_c)}(W_4) & = & (1 \times 0.6 + 0 \times 0.4) \times p_{s_c}(W_1) + 0 \times 1 \times p_{s_c}(W_2) \\ & \simeq & 0.6 \times 0.36 \simeq 0.22, \\ p_{(table2up,s_c)}(W_2) & = & (0 \times 0.6 + 1 \times 0.4) \times p_{s_c}(W_1) + 1 \times 1 \times p_{s_c}(W_2) \\ & \simeq & 0.4 \times 0.36 + 1 \times 0.64 \simeq 0.78. \end{array}$$

Auf diese Weise können MW-Pläne mit Übergangswahrscheinlichkeiten etikettiert werden. Abbildung 15 gibt diese Wahrscheinlichkeiten für den Plan Π_c aus Abbildung 11 an.

MW-Pläne haben im allgemeinen eine solche Gestalt, daß es darin Schleifen gibt, die im Prinzip unendlich oft durchlaufen werden können, wie beispielsweise der Operator **DREHE** in Π_c. Es ist daher im Prinzip nicht trivial anzugeben, mit welcher Wahrscheinlichkeit vom Startknoten eines Plans aus ein bestimmter Zielknoten erreicht werden kann, wobei auf dem Weg dazwischen solche Schleifen durchlaufen werden können. Doch es gibt ein recht einfaches mathematisches Modell dafür: stationäre Markow-Ketten. Wie man MW-Pläne darauf abbildet, ist eine technische Frage, die hier nicht interessieren soll; ich verweise auf [Hertzberg und Thiébaux, 1993].

Wir haben damit alles Material zusammen, um den Grad an Korrektheit und Vollständigkeit eines MW-Plans (KV-Grad) genauso wie oben intuitiv beschrieben zu definieren. Danach wird gezeigt, daß der KV-Grad eine Planbewertungsfunktion darstellt.

Definition 26 (KV-Grad) *Seien* Π *ein* MW*-Plan für* $\Psi = \langle s, O, f, C, B\rangle$, P *das probabilistische Hintergrundwissen. Für alle* T*-Knoten* τ *in* Π *sei*

$$Succ(\tau, \Pi) = \{w \mid \textit{es gibt einen Nachfolger } \omega \textit{ von } \tau \textit{ in } \Pi \textit{ etikettiert mit } w\}.$$

KV-Grad der Wurzel. *Der KV-Grad von* START *in* Π *bzgl.* Ψ *ist*

$$\mathrm{ccd}_\Psi(\textsc{start}, \Pi) = \frac{p_s(Succ(\textsc{start}, \Pi))}{card(Succ(\textsc{start}, \Pi) \setminus \mathrm{Poss}_K(s)) + 1}$$

KV-Grad der anderen T**-Knoten.** *Sei* τ *mit* o *etikettiert ein von* START *verschiedener* T*-Knoten, und sei* ω *etikettiert mit* w *sein Vorgänger in* Π*. Der KV-Grad von* τ *in* Π *bzgl.* Ψ *ist*

$$\mathrm{ccd}_\Psi(\tau, \Pi) = \frac{p_{(o,w)}(Succ(\tau, \Pi))}{card(Succ(\tau, \Pi) \setminus r(o, w)) + 1}$$

KV-Grad des Plans. *Bezeichne für einen* T*-Knoten* τ *in* Π *der Ausdruck* $\pi(X_t = \tau \mid X_0 = \textsc{start})$ *die Wahrscheinlichkeit dafür, daß* τ *der* t*-te angewendete Operator bei einer Planausführung beginnend bei* START *ist; für einen Blattknoten* ω *in* Π *bezeichne* $\pi(X_t = \omega \mid X_0 = \textsc{start})$ *die Wahrscheinlichkeit, daß* ω *nach* t *oder weniger Operatoranwendungen aus* START *heraus erreicht ist.*[29] *Der KV-Grad von* Π *bzgl.* Ψ *ist*

$$\mathrm{ccd}_\Psi(\Pi) = c_1 \cdot ccd_{T-Knoten} + c_2 \cdot ccd_{Blatt},$$

wobei c_1, c_2 *positive reelle Konstanten sind, so daß* $c_1 + c_2 = 1$ *und*

$$ccd_{T-Knoten} = \frac{\displaystyle\sum_{\tau \in \mathcal{T}} \left(\mathrm{ccd}_\Psi(\tau, \Pi) \cdot \sum_{t=0}^{\infty} \pi(X_t = \tau \mid X_0 = \textsc{start}) \right)}{\displaystyle\sum_{\tau \in \mathcal{T}} \sum_{t=0}^{\infty} \pi(X_t = \tau \mid X_0 = \textsc{start})}$$

$$ccd_{Blatt} = \sum_{\omega\text{: Blatt, in dem } f \text{ gilt}} \left(\lim_{t \to \infty} \pi(X_t = \omega \mid X_0 = \textsc{start}) \right)$$

Aus der Konstruktion der ccd-Funktion ergibt sich unmittelbar der folgende Sachverhalt:

Lemma 1 ccd_Ψ *ist eine Planbewertungsfunktion für alle* MW*-Problembeschreibungen* Ψ*.*

[29] Diese Konzepte sind in [Hertzberg und Thiébaux, 1993] über stationäre Markow-Ketten definiert.

Damit ist weiterhin klar, daß die Prozedur MWPU aus Abbildung 12 begrenzt-korrekt ist, wenn man für die Funktion QUALITY(Ψ) die Funktion ccd_Ψ einsetzt. Allerdings kann man dieses Ergebnis noch auf triviale Weise etwas verallgemeinern zu der folgenden Aussage:

Lemma 2 (Begrenzt-Korrektheit von MWPU) MWPU *ist begrenzt-korrekt, wenn die Funktion* QUALITY(Π) *ordnungserhaltend für alle* MW*-Problembeschreibungen* Ψ *ist im folgenden Sinn:*

1. *wenn* $\mathrm{QUALITY}(\Pi_1) > \mathrm{QUALITY}(\Pi_2)$, *dann* $\mathrm{ccd}_\Psi(\Pi_1) > \mathrm{ccd}_\Psi(\Pi_2)$, *und*
2. *wenn es einen korrekten und vollständigen Plan für* Ψ *gibt, dann gibt es einen Plan* Π, *für den* $\mathrm{ccd}_\Psi(\Pi) = 1$, *und für alle* Π': *Wenn* $\mathrm{ccd}_\Psi(\Pi') < 1$, *dann* $\mathrm{QUALITY}(\Pi) > \mathrm{QUALITY}(\Pi')$

In einer so definierten Qualitätsfunktion ist der KV-Grad eines Plans das alles entscheidende Qualitätsmerkmal. Das mag praktisch oft unrealistisch sein, wenn beispielsweise ein Nutzen von Zielknoten definiert ist (nicht jede Zielwelt muß gleich „erwünscht“ sein wie in Definition 26 impliziert) und dieser Nutzen in die Bewertung eines Plans eingehen soll; oder wenn Operatorkosten definiert sind, die als negativer Nutzen in die Planbewertung eingerechnet werden; oder wenn gar der Nutzen eines Plans abhängig ist von der Zeit, die man gebraucht hat, ihn zu finden, Rechenzeit also als Kosten eines Plans zu Buche schlagen soll. Alle solche Überlegungen können dazu führen, eine Planbewertungsfunktion wie den KV-Grad für MW-Pläne nicht zum einzigen Qualitätsmaß eines Planes zu machen und damit beispielsweise die Forderung aufzugeben, ein Planer habe über die Zeit nur Pläne mit monoton wachsender Planbewertungsfunktion zurückzugeben (sondern lieber mit wachsendem „freiem“ Qualitätsmaß). Anliegen dieses Abschnitts war es jedoch, zu zeigen, daß auch „unfertige“ Pläne an Operatorkalkülen gemessen werden können und wie diese Messung in ein Qualitätsmaß für Pläne eingehen kann.

6.3 Diskussion: Wozu Begrenzt-Korrektheit?

Ich will schließen mit einer kurzen Diskussion der Frage, wofür das Konzept Begrenzt-Korrektheit überhaupt taugt. Hintergrund war der Wunsch, die Bedeutung eines Plans festzumachen an einem formal definierten Kalkül zur Berechnung der Effekte von Operatoren. Bereits hierüber kann man streiten: Zum Beispiel haben die Implementatoren von STRIPS ihren Roboter Shakey [Nilsson, 1984] auch ohne formalen Operatorkalkül Pläne ausführen lassen können, und meines Wissens haben weder der Editor, noch das Textsatzprogramm, die ich für diesen Text verwende, eine formale Semantik – und doch kann ich sie beide vernünftig verwenden.

Die Frage, ob automatisch generierte Pläne einer irgendwie gearteten formalen Semantik entsprechen sollen, ist letztlich vergleichbar mit der Frage, ob Programme einer formalen Spezifikation beweisbar entsprechen sollen. Hier kann man unterschiedlicher Meinung sein, und meine Meinung dazu ist die: Im derzeitigen Stand der Informatik mag es praktisch in Ordnung sein, Programme abzuliefern, die eine intuitiv angegebene Funktion intuitiv plausibel erfüllen. Aber dabei sollte es nicht in alle Ewigkeit bleiben. Dasselbe gilt für automatisch generierte Pläne: Auf lange Sicht sollte an Pläne ein formales Korrektheitskriterium angelegt werden können, das angibt, inwieweit die formale Zielspezifikation durch einen Plan erfüllt wird.

Bei der Korrektheit von Plänen kommt gegenüber der von Programmen noch ein Weiteres hinzu: Die Bedeutung der Befehle einer Programmiersprache, also beispielsweise die Bedeutung einer Zuweisung in einer imperativen Sprache wie PASCAL, ist klar definiert. Die „Bedeutung“ von Operatoren ist weitaus weniger klar. Sie muß aber mindestens dann auch im Detail bekannt sein, wenn man einen Anwendungsbereich modelliert, also die Wirkungen von Handlungen als Nachbedingungen von Operatoren formuliert. Kennt der Bereichsmodellierer dann die Bausteine seiner Sprache nicht genau, kann er ungewollt und unbewußt falsch modellieren. Beispielsweise war vor Lifschitz' [1987] STRIPS-Semantik der Unterschied zwischen wesentlichen und nicht-wesentlichen Formeln für STRIPS-Operatoren nicht bekannt oder doch nicht im allgemeinen Bewußtsein; im einfachsten Fall konnte das zu umständlichen Modellierungen von Anwendungen mit STRIPS-Operatoren führen (wenn man für sich selber „im Kopf“ alle Formeln als wesentlich angesehen und einzeln behandelt hat), und im schlimmsten Fall zu falschen Modellierungen (wenn man auf die natürlichsprachliche Beschreibung der Wirkungen von STRIPS-Operatoren gebaut hat).

Die „harte Linie“ der Korrektheit wäre nun, ausschließlich strikt korrekte Planer zuzulassen, wie in Definition 20 angegeben. In diesem Fall ist es konsequent, Planen als das Finden eines konstruktiven Beweises im zugrundeliegenden Operatorkalkül aufzufassen, daß ein Plan die Startsituation in eine Zielsituation überführt. Das ist genau die Philosophie des *deduktiven* Planens [Biundo, 1994].

Aus zwei Gründen halte ich diese „harte Linie“ *im allgemeinen* nicht für sinnvoll.

- Zum einen folgt aus der Forderung nach strikter Planerkorrektheit nicht, daß der Planer auch deduktiv arbeiten muß. Er kann seine Pläne finden wie er mag – *deduktiv* muß anschließend nur die Korrektheit und Vollständigkeit seiner Ergebnis-Pläne nachgewiesen werden. In Anwendungen und Operatorkalkülen, wo ein konstruktives Beweisverfahren praktisch und im Schnitt effektiver ist als alle anderen beliebigen heuristischen Verfahren, bestimmt dieses Beweisverfahren den Planer der Wahl; in allen anderen Fällen wäre es ungeschickt, sich nur aus Prinzip auf ein deduktives Verfahren festzulegen.

- Zum zweiten soll man bei Plänen (wie auch bei Programmen) einen formalen Beweis der Korrektheit und Vollständigkeit auch nicht überbewerten. Dieser Beweis kann immer nur relativ sein

  - zur Korrektheit der Bereichsmodellierung: Hat der Modellierer den Bereich intuitiv falsch modelliert, hilft der schönste Korrektheitsbeweis für Pläne nicht;
  - zur Sicherheit und Vollständigkeit der Information: In den praktisch sehr häufigen Fällen, wo die Bereichsinformation zur Planungs- und zur Ausführungszeit unsicher und unvollständig ist, vielleicht sogar sich ständig ändert, macht es wenig Sinn, auf allerkorrektester Verarbeitung dieser Information zu beharren. Wenn man das effizient und rechtzeitig kann, ist es schön, und dann sollte man's auch tun; andernfalls wird die Verarbeitungsungenauigkeit oft klein sein gegen die Modellierungsungenauigkeit.

Ein Operatorkalkül ist eine Meßlatte für die Pläne, die ein Planer erzeugt, kein Prokrustes-Bett, in das Planungsverfahren hineingezwungen werden müssen. Langfristig braucht man Kalküle und Planer derart, daß Pläne „meßbar“ korrekt sind – nicht mehr und nicht weniger. Begrenzt-Korrektheit ist *ein* Schritt dahin, höchstwahrscheinlich nicht der letzte, beste und endgültige.

Literatur

[Agre und Chapman, 1987] P. Agre und D. Chapman. Pengi: An implementation of a theory of action. In *Proc. AAAI-87*, S. 268–272, San Mateo, CA, 1987. Morgan Kaufmann.

[Allen, 1991] J. Allen. Temporal Reasoning and Planning. In J. Allen, H. Kautz, R. Pelavin und J. Tenenberg (Hrsg.), *Reasoning about Plans*, Kapitel 1, S. 1–68. Morgan Kaufmann, 1991.

[Alterman, 1986] R. Alterman. An Adaptive Planner. In *Proc. AAAI-86*, S. 65–69, San Mateo, CA, 1986. Morgan Kaufmann.

[Biundo, 1994] S. Biundo. Present-Day Deductive Planning. In C. Bäckström und E. Sandewall (Hrsg.), *Current Trends in AI Planning. EWSP'93 – 2nd European Workshop on Planning*, S. 1–5. IOS Press, 1994.

[Bond und Gasser, 1988] A. H. Bond und L. Gasser (Hrsg.). *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann, San Mateo, CA, 1988.

[Brewka und Hertzberg, 1993] G. Brewka und J. Hertzberg. How to Do Things with Worlds: On Formalizing Actions and Plans. *J. Logic and Computation*, 3(5):517–532, 1993.

[Brewka, 1993] G. Brewka. Nichtmonotones Schließen. In G. Görz (Hrsg.), *Künstliche Intelligenz*, S. 55–85. Addison Wesley, Bonn, 1993.

[Bylander, 1991] T. Bylander. Complexity Results for Planning. In *Proc. IJCAI-91*, S. 274–279, San Mateo, CA, 1991. Morgan Kaufmann.

[Chapman, 1987] D. Chapman. Planning for Conjunctive Goals. *J. Art. Intell.*, 32:333–377, 1987.

[Cordier und Siegel, 1992] M.O. Cordier und P. Siegel. A Temporal Revision Model for Reasoning about World Change. In B. Nebel, C. Rich und W. Swartout (Hrsg.), *Principles of Knowledge Representation and Reasoning (KR'92)*, S. 732–739. Morgan Kaufmann, 1992.

[Dean und Boddy, 1988] T.L. Dean und M. Boddy. An Analysis of Time-dependent Planning. In *Proc. AAAI-88*, S. 49–54, 1988.

[Dean und McDermott, 1987] T.L. Dean und D.V. McDermott. Temporal Data Base Management. *J. Art. Intell.*, 32:1–55, 1987.

[Dean und Wellman, 1991] T.L. Dean und M.P. Wellman. *Planning and Control*. Morgan Kaufmann, San Mateo, CA, 1991.

[Dörner, 1989] D. Dörner. *Die Logik des Mißlingens. Strategisches Denken in komplexen Situationen.* Rowohlt, Reinbek, 1989.

[Drummond, 1989] M. Drummond. Situated Control Rules. In *Proc. KR-89*, S. 103–113, 1989.

[Feldman und Sproull, 1977] J.A. Feldman und R.F. Sproull. Decision Theory and Artificial Intelligence II: The Hungry Monkey. *Cogn. Sci.*, 1:158, 1977.

[Fikes *et al.*, 1972] R.E. Fikes, P.E. Hart und N.J. Nilsson. Learning and Executing Generalized Robot Plans. *J. Art. Intell.*, 3:251–288, 1972.

[Fikes und Nilsson, 1971] R.E. Fikes und N.J. Nilsson. STRIPS: A New Approach to Theorem Proving in Problem Solving. *J. Art. Intell.*, 2:189–208, 1971.

[Finger, 1987] J.J. Finger. *Exploiting Constraints in Design Synthesis.* Dissertation, Stanford University, 1987.

[Firby, 1992] R.J. Firby. Building Symbolic Primitives with Continuous Control Routines. In *[Hendler, 1992]*, S. 62–69, 1992.

[Fox und Smith, 1984] M.S. Fox und S.F. Smith. ISIS—A Knowledge-based System for Factory Scheduling. *Expert Systems*, 1(1):25–49, 1984.

[Georgeff und Lansky, 1987] M.P. Georgeff und A.L. Lansky. Reactive Reasoning and Planning. In *Proc. AAAI-87*, S. 677–682, 1987.

[Ginsberg und Smith, 1988a] M.L. Ginsberg und D.E. Smith. Reasoning about Action I: A Possible Worlds Approach. *J. Art. Intell.*, 35:165–195, 1988.

[Ginsberg und Smith, 1988b] M.L. Ginsberg und D.E. Smith. Reasoning about Action II: The Qualification Problem. *J. Art. Intell.*, 35:311–342, 1988.

[Ginsberg, 1989] M.L. Ginsberg. Universal Planning: An (Almost) Universally Bad Idea. *AI Magazine*, 10(4 (Winter)):40–44, 1989.

[Ginsberg, 1993] M.L. Ginsberg. *Essentials of Artificial Intelligence.* Morgan Kaufmann, San Mateo, CA, 1993.

[Green, 1969] C. Green. Application of Theorem Proving to Problem Solving. In *Proc. IJCAI-69*, S. 219–239. Morgan Kaufmann, 1969.

[Hammond, 1989] K. Hammond. *Case Based Planning. Viewing Planning as a Memory Task.* Perspectives in Artificial Intelligence Series. Academic Press, Boston, 1989.

[Hanks und McDermott, 1986] S. Hanks und D. McDermott. Default Reasoning, Nonmonotonic Logics, and the Frame Problem. In *Proc. AAAI-86*, S. 328–333, 1986.

[Hendler *et al.*, 1990] J. Hendler, A. Tate und M. Drummond. AI Planning: Systems and Techniques. *AI Magazine*, 11(2):61–77, 1990.

[Hendler, 1992] J. Hendler (Hrsg.). *Artificial Intelligence Planning Systems: Proc. of the First International Conference (AIPS92)*, San Mateo, CA, 1992. Morgan Kaufmann.

[Hertzberg und Horz, 1989] J. Hertzberg und A. Horz. Towards a Theory of Conflict Detection and Resolution in Nonlinear Plans. In *Proc. IJCAI-89*, S. 937–942, 1989.

[Hertzberg und Thiébaux, 1993] J. Hertzberg und S. Thiébaux. Turning an Action Formalism into a Planner—A Case Study. Technical Report 93-033, ICSI, Berkeley, Kalifornien, 1993. angenommen zur Veröffentlichung bei *J. Logic and Computation.*

[Hertzberg, 1989] J. Hertzberg. *Planen. Einführung in die Planerstellungsmethoden der Künstlichen Intelligenz.* BI Wissenschaftsverlag, Mannheim u.a., 1989.

[Hertzberg, 1990] J. Hertzberg. Revising Planning Goals: On Concept and Implementation. In N. Kratz, A. Günter und J. Hertzberg (Hrsg.), *Beiträge zum 4. Workshop Planen und Konfigurieren.* FAW Ulm, Bericht Nr. FAW-B-90008, 1990.

[Hertzberg, 1993] J. Hertzberg. KI-Handlungsplanung – Woran wir arbeiten, und woran wir arbeiten sollten. In O. Herzog, Th. Christaller und D. Schütt (Hrsg.), *Grundlagen und Anwendungen der Künstlichen Intelligenz. 17. Fachtagung für Künstliche Intelligenz (KI'93)*, S. 3–27, 1993.

[Hoffmann, 1991] A. Hoffmann. Die Theorie des Lernbaren – ein Überblick. *KI*, 5(1):7–11, 1991.

[Krebsbach *et al.*, 1992] K. Krebsbach, D. Olawsky und M. Gini. An Empirical Study of Sensing and Defaulting in Planning. In *[Hendler, 1992]*, S. 136–144, 1992.

[Lifschitz, 1987] V. Lifschitz. On the Semantics of STRIPS. In M.P. Georgeff und A.L. Lansky (Hrsg.), *Proc. 1986 Workshop Reasoning about Actions and Plans, Timberline, OR*, S. 1–9, Los Altos, 1987. Morgan Kaufmann.

[Manna und Waldinger, 1987] Z. Manna und R. Waldinger. A Theory of Plans. In M.P. Georgeff und A.L. Lansky (Hrsg.), *Proc. 1986 Workshop Reasoning about Actions and Plans, Timberline, OR*, San Mateo, 1987. Morgan Kaufmann.

[McAllester und Rosenblitt, 1991] D. McAllester und D. Rosenblitt. Systematic Nonlinear Planning. In *Proc. AAAI-91*, S. 634–639, 1991.

[McCarthy und Hayes, 1969] J. McCarthy und P. Hayes. Some Philosophical Problems from the Standpoint of Artificial Intelligence. *Machine Intelligence*, 4:463–507, 1969.

[McDermott, 1992] D. McDermott. Robot Planning. *AI Magazine*, 13(2):55–79, 1992.

[Minton, 1985] S. Minton. Selectively Generalizing Plans for Problem-Solving. In *Proc. IJCAI-85*, S. 596–599, 1985.

[Nilsson, 1982] N.J. Nilsson. *Principles of Artificial Intelligence.* Springer Verlag, Berlin, 1982.

[Nilsson, 1984] N.J. Nilsson. Shakey the robot. Technical Note 323, SRI International, April 1984.

[Pednault, 1988] E.P.D. Pednault. Synthesizing Plans that Contain Actions with Context-Dependent Effects. *J. Comp. Intell.*, 4:356–372, 1988.

[Pelavin, 1991] R. Pelavin. Planning with Simultaneous Actions and External Events. In J. Allen, H. Kautz, R. Pelavin und J. Tenenberg (Hrsg.), *Reasoning about Plans*, Kapitel 3, S. 127–211. Morgan Kaufmann, 1991.

[Peot und Smith, 1992] M.A. Peot und D.E. Smith. Conditional Nonlinear Planning. In J. Hendler (Hrsg.), *Artificial Intelligence Planning Systems: Proc. of the First International Conference (AIPS92)*, S. 189–197, San Mateo, CA, 1992. Morgan Kaufmann.

[Pylyshyn, 1987] Z.W. Pylyshyn (Hrsg.). *The Robot's Dilemma: The Frame Problem in Artificial Intelligence.* Ablex, Norwood, NJ, 1987.

[Reiter, 1980] R. Reiter. A Logic for Default Reasoning. *J. Artificial Intelligence*, 13:81–132, 1980.

[Ringle, 1993] M.D. Ringle (Hrsg.). *Special Issue: Situated Action*, Band 17(1) der *Cogn. Sci.* 1993.

[Rosenschein, 1981] J.S. Rosenschein. Plan Synthesis: A Logical Perspective. In *Proc. IJCAI-81*, S. 331–337, 1981.

[Russell und Wefald, 1991] S. Russell und E. Wefald. *Do the Right Thing. Studies in Limited Rationality.* MIT Press, Cambridge, Massachusetts, 1991.

[Russell und Zilberstein, 1991] S.J. Russell und S. Zilberstein. Composing Real-Time Systems. In *Proc. IJCAI-91*, S. 212–217, San Mateo, CA, 1991. Morgan Kaufmann.

[Sacerdoti, 1974] E.D. Sacerdoti. Planning in a Hierarchy of Abstraction Spaces. *J. Art. Intell.*, 5(2), 1974.

[Sacerdoti, 1977] E.D. Sacerdoti. *A Structure for Plans and Behavior.* Elsevier/North Holland, 1977.

[Sandewall, 1992] E. Sandewall. Features and Fluents. A Systematic Approach to the Representation of Knowledge about Dynamical Systems. LiTH-IDA-R-92-30, Dept. of Computer and Information Sciences, Linköping University, 1992.

[Shoham und McDermott, 1988] Y. Shoham und D. McDermott. Problems in Formal Temporal Reasoning. *J. Artificial Intelligence*, 36:49–61, 1988.

[Shoham, 1988] Y. Shoham. *Reasoning about Change. Time and Causation from the Standpoint of Artificial Intelligence.* The MIT Press series in Artificial Intelligence. MIT Press, Cambridge, Massachusetts, 1988.

[Simon, 1976] H. A. Simon. *Administrative Behavior. A Study of Decision-Making Processes in Administrative Organizations.* Free Press, New York, 3 Auflage, 1976. Original: New York (Macmillan) 1947.

[Sussman, 1975] G. Sussman. *A Computer Model of Skill Acquisition.* Elsevier, New York, 1975.

[Tate, 1977] A. Tate. Generating Project Networks. In *Proc. IJCAI-77*, S. 888–893, San Mateo, CA, 1977. Morgan Kaufmann.

[Tenenberg, 1991] J.D. Tenenberg. Abstraction in Planning. In J. Allen, H. Kautz, R. Pelavin und J. Tenenberg (Hrsg.), *Reasoning about Plans*, Kapitel 4, S. 213–283. Morgan Kaufmann, 1991.

[Vere, 1983] S.A. Vere. Planning in Time: Windows and Durations for Activities and Goals. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-5:246–267, 1983.

[von Martial, 1993] F. von Martial. Planen in Multi-Agenten Systemen. In J. Müller (Hrsg.), *Verteilte Künstliche Intelligenz: Methoden und Anwendungen*, Kapitel 4. BI-Wissenschaftsverlag (Reihe Informatik), Mannheim u.a., 1993.

[Walther, 1993] Ch. Walther. Automatisches Beweisen. In G. Görz (Hrsg.), *Künstliche Intelligenz*, S. 205–246. Addison Wesley, Bonn, 1993.

[Wilkins, 1988] D. Wilkins. *Practical Planning. Extending the Classical AI Planning Paradigm.* Morgan Kaufmann, San Mateo, CA, 1988.

[Winslett, 1988] M. Winslett. Reasoning about Action Using a Possible Models Approach. In *Proc. AAAI-88*, S. 89–93, 1988.