Projektarbeit

Das Modal Truth Criterion für partiell geordnete Pläne und seine Realisierung im Planungswerkzeug CAPlan

Thomas Roth-Berghofer

Juli 1995

Betreuung: Prof. Dr. Michael M. Richter
Dipl. Inform. Frank Weberskirch

Arbeitsgruppe Künstliche Intelligenz
— Expertensysteme —
Prof. Dr. Michael M. Richter

Universität Kaiserslautern
Fachbereich Informatik

Inhaltsverzeichnis

Zusammenfassung

David Chapman hat 1987 mit seinem Artikel *Planning for Conjunctive Goals* einen wichtigen Schritt in Richtung der Formalisierung von partiell ordnender Planung und ihrem Verständnis gemacht. Grundlegendes Konzept von David Chapman ist die Idee modaler Wahrheit in Plänen; sein *Modal Truth Criterion* (MTC) macht Aussagen über die Gültigkeit einer Aussage in einem partiell geordneten Plan.

Kambhampati und Nau zeigten mit ihrem Papier *On the Nature of Modal Truth Criteria in Planning* einige wesentliche Mängel bzw. begriffliche Ungenauigkeiten an Chapmans Kriterium auf und machten mit ihrem *Modal Conditional Truth Criterion* Vorschläge für eine Korrektur. Insbesondere bekam die Frage nach der Ausführbarkeit eines Planes hier ein größeres Gewicht.

Ziel dieser Projektarbeit ist es zunächst, das MTC von Chapman sowie die Modifikation darzustellen und, als praktische Aufgabe, die Realisierung des MTC für den Causal Link Planer CAPlan, basierend auf der bestehenden Implementation des Systems.

Kapitel 1

Einleitung

1.1 Motivation

Planungssysteme mit partiell geordneter Planrepräsentation benötigen Kriterien, um zu jedem beliebigen Zeitpunkt den Wahrheitsgehalt eines Prädikates feststellen zu können. David Chapman entwickelte das Modal Truth Criterion (im folgenden MTC abgekürzt)[Cha87], das Aussagen darüber macht, ob ein gewähltes Prädikat an einer bestimmten Stelle im Plan notwendigerweise oder nur möglicherweise wahr ist.

Chapman trug in [Cha87] verschiedenste Techniken, die in einzelnen anderen Planungssystemen bereits eingesetzt worden waren, zusammen und konstruierte ein präzise definiertes System, das klare Aussagen über seine Einsatzbedingungen zuläßt. In seinem Planungsprogramm TWEAK setzte er das MTC, das nun die verschiedenen Techniken in sich vereinigte, als Planungsalgorithmus ein (vgl. [Cha85, Cha87]).

Die Formalisierung Chapmans war jedoch nicht so präzise, wie er glaubte. Auf verschiedene Ungenauigkeiten wurde bereits an anderer Stelle hingewiesen (z.B. bzgl. der Definition von Situationen in [YT90] oder bzgl. Chapmans zweitem Unentscheidbarkeitstheorem in [ENS91, ENS92]).

Subbarao Kambhampati und Dana S. Nau behandeln in [KN94, KN95] das Problem der möglichen Wahrheit eines Prädikats. Chapmans Aussage darüber ist fehlerhaft. Kambhampati und Nau zeigen dies, analysieren aber auch die Gründe dafür und bieten eine Erweiterung des MTC an, die modale bedingte Wahrheit (modal conditional truth), für die sie die Korrektheit und Berechenbarkeit in polynomialer Zeit nachweisen.

Diese Projektarbeit stellt das MTC im Sinne von Chapman und die Erweiterung nach Kambhampati und Nau vor und beschreibt kurz die Implementierung dieser Konzepte im SNLP-basierten Planungssystem CAPlan.

1.2 Übersicht über die Arbeit

Die vorliegende Arbeit gliedert sich wie folgt:

- Kapitel 2 dient zur Klärung der Hauptbegriffe und der Einführung einer Terminologie für diese Arbeit. Unter anderem werden die Begriffe *Modallogik*, *möglicherweise wahr* und *notwendigerweise wahr* erläutert, sowie *Planung* und *nichtlineares* Planen.

- Kapitel 3 stellt das modale Wahrheitskriterium (modal truth criterion) von David Chapman vor. Dieses Kriterium wurde als Planungsverfahren im System TWEAK eingesetzt.
- Kapitel 4 befaßt sich mit dem modalen bedingten Wahrheitskriterium (modal conditional truth criterion), das nicht die Ausführbarkeit des zugehörigen Planes verlangt, die Chapmans MTC implizit voraussetzt. Kambhampati und Nau gewähren außerdem Einblick in die Komplexität der einzelnen Kriterien und stellen Ungenauigkeiten Chapmans richtig.
- In Kapitel 5 wird die Implementation des MTC im Planungssystem CAPlan kurz vorgestellt.

Die in Kapitel 5 vorgestellte Erweiterung von CAPlan wurde, wie das Gesamtsystem selbst, in SMALLTALK-80, RELEASE 4.1, auf den Rechnern der AG Richter realisiert. Das Verständnis der Implementation setzt daher gründliche Kenntnisse dieser Sprache und des objektorientierten Ansatzes voraus. Empfohlene Literatur dafür ist z.B. [GR83].

Kapitel 2

Grundlagen und Terminologie

2.1 Grundlagen

2.1.1 Modale Logik

Um das Verständnis dieser Projektarbeit zu erleichtern, soll hier kurz die wesentliche Idee der Modallogik und modaler Ausdrucksweisen wiedergegeben werden, da sie beim MTC eine zentrale Rolle spielen. Die Definitionen stammen aus [Ric92].

Die klassische Logik beschreibt Eigenschaften jeweils eines Modells. Im Gegensatz dazu betrachtet die Modallogik ein ganzes System möglicher Welten. Diese Welten sind nicht unabhängig voneinander. Oft gibt es durch Handlungen, die die Welt verändern, oder durch das Fortschreiten der Zeit Übergangsmöglichkeiten von einer Welt zur anderen. Dargestellt wird dies meist durch Graphen.

In einem solchen Universum mit möglichen Welten kann man sich über Möglichkeiten und Notwendigkeiten unterhalten. Die Möglichkeit entspricht dabei im Prinzip einer Existenzquantifizierung über die Welten, und die Notwendigkeit bedeutet eine All-Quantifizierung. Eine dazu erweiterte prädikatenlogische Sprache sieht so aus:

Modallogische Sprache: Eine *modallogische Sprache ML* entsteht aus einer prädikatenlogischen Sprache PL durch Hinzunahme zweier logischer Zeichen $\Box$ und $\Diamond$ zur Menge der logischen Zeichen dieser Sprache. Sie werden wie das einstellige logische Zeichen $\neg$ verwendet, also: Mit ϕ sind auch $\Box\phi$ (sprich: ϕ gilt *notwendigerweise*) sowie $\Diamond\phi$ (sprich: ϕ gilt *möglicherweise*) Formeln.

Die allgemein zugrundegelegte Auffassung ist nun, daß die Modaloperatoren $\Box$ und $\Diamond$ im selben Zusammenhang wie die Quantoren $\forall$ und $\exists$ stehen, d.h. $\Box$ wird nur als Grundzeichen angesehen und $\Diamond$ als Abkürzung für $\neg\Box\neg$ aufgefaßt. Dieser Zusammenhang wird im folgenden auch *modale Dualität* genannt.

2.1.2 Planung und nichtlineares Planen

Hier sollen die Begriffe *Planung* und *nichtlineares Planen* in Anlehnung an [Web94] kurz erläutert werden.

Unter *Planung* versteht man im Gegensatz zur Diagnose, die hautpsächlich Analyseaufgaben betrachtet, vornehmlich die Lösung von Syntheseproblemen. Die Lösung einer Planungsaufgabe soll mittels vorgegebener Elementarbausteine erstellt, synthetisiert, werden und dabei gewisse Kriterien und Wertvorstellungen erfüllen. Eine verfeinerte Sichtweise unterscheidet die Planung noch einmal in *Konfiguration* und *Aktionsplanung*. Bei der Konfiguration stellt sich die Frage, aus welchen primitiven Komponenten ein komplexes Objekt zusammengesetzt werden muß, um bestimmte Anforderungen zu erfüllen (z.B. Konfiguration eines Fahrrades aus Rahmen, Lenker, etc.). In der Aktionsplanung besteht das zu planende Objekt aus einer Reihe von Aktionen, die eine Anfangssituation in eine Zielsituation überführen (z.B. Roboteraktionen). In dieser Projektarbeit geht es nur um die Planung im Sinne der Aktionsplanung.

Der Begriff *nichtlineare Planung* wurde in der Literatur oft mißverstanden und fälschlicherweise mit einer partiell geordneten Planrepräsentation gleichgesetzt, obwohl damit zwei verschiedene Aspekte vermischt wurden. So wurde z.B. das System SIPE [Wil88] oft als nichtlineares Planungssystem bezeichnet, obwohl es lediglich eine partielle Ordnung im Plan zuläßt und nur serialisierbare Mengen von Teilzielen lösen kann. Hier sei daher die in [Web94] beschriebene Bedeutung vorausgesetzt:

Nichtlineare Planung: *Nichtlineare Planung* unterscheidet sich vom linearen Planen dadurch, daß sie auch für nicht-serialisierbare[1] Mengen von Teilzielen eine Lösung findet.

2.2 Terminologie

In diesem Kapitel werden einige Notationen und Begriffe zur Erklärung der Konzepte von Chapman sowie von Kambhampati und Nau vorgestellt. Auf Unterschiede wird an den entsprechenden Stellen hingewiesen. Die Definitionen sind zum größten Teil [KN95] entnommen.

Die Planungssprache $\mathcal{L}$ ist eine beliebige *funktionsfreie* Sprache erster Ordnung. Da $\mathcal{L}$ funktionsfrei ist, kann jeder Term nur entweder ein Variablen- oder ein Konstantensymbol sein (Chapman fordert hier unendlich viele Konstanten- und Variablensymbole). Ein *Atom* ist ein Prädikatensymbol, gefolgt von einer Liste von Termen, einem *Literal* oder einer *Aussage*. Ein Literal ist ein Atom oder dessen Negation. Eine Aussage ist ein nullstelliges Atom.[2]

Planschritt und partielle Ordnung. Ein *Schritt* bezeichnet eine Aktion und ist ein Tripel $s = (name(s), pre(s), post(s))$, wobei $name(s)$ eine Konstante ist, die *Name von s* heißt,[3] und $pre(s)$ und $post(s)$ Mengen von Literalen sind, die *Vor-* und *Nachbedingungen* genannt werden. Die Menge der Vorbedingungen muß vor, die Menge der Nachbedingungen muß nach der Ausführung des Schrittes gelten.

Wenn $\mathcal{S}$ eine Menge von Schritten ist, dann ist ein *Ordnungs-Constraint* auf $\mathcal{S}$ ein syntaktischer Ausdruck der Form '$s \prec t$' (s ist Vorgänger von t), mit $s, t \in \mathcal{S}$. Er ordnet Schritt s vor Schritt t. Daraus ergibt sich eine partielle Ordnung der Schritte. Wenn $\mathcal{O}$ eine Menge von Ordnungs-Constraints auf $\mathcal{S}$ und $\prec$ eine totale Ordnung auf $\mathcal{S}$ ist, dann erfüllt $\prec$ die Menge $\mathcal{O}$, wenn für jeden syntaktischen Ausdruck '$s \prec t$' in $\mathcal{O}$ gilt: $s \prec t$.

[1] Eine Menge von Teilzielen ist nicht-serialisierbar, wenn eine Verzahnung der Teilpläne notwendig ist, um noch eine korrekte Lösung zu finden.

[2] Dies nennt Chapman ein Literal.

[3] Im folgenden werden s und $name(s)$ synonym benutzt.

Übereinstimmung. Übereinstimmung (engl. codesignation) ist eine Äquivalenzrelation auf Variablen und Konstanten. Sei $\mathcal{T}$ eine endliche Menge von Termen. Ein *Bindungs-Constraint* auf $\mathcal{T}$ ist ein syntaktischer Ausdruck der Form '$t \approx u$' oder '$t \not\approx u$', mit $t, u \in \mathcal{T}$.

Sei $\mathcal{B}$ eine Menge von Bindungs-Constraints auf $\mathcal{T}$ und θ eine *Grundsubstitution* für $\mathcal{T}$ (z.B. eine Substitution, die jeder Variablen in $\mathcal{T}$ einen Grundterm zuweist). Dann erfüllt θ die Menge $\mathcal{B}$, wenn $t\theta \approx u\theta$ gilt für jeden syntaktischen Ausdruck '$t \approx u$' in $\mathcal{B}$, und $t\theta \not\approx u\theta$ gilt für jeden syntaktischen Ausdruck '$t \not\approx u$' in $\mathcal{B}$.

$\mathcal{B}$ ist konsistent, wenn es mindestens eine Grundsubstitution θ gibt, die $\mathcal{B}$ erfüllt. Wenn $t\theta \approx u\theta$ für jedes θ gilt, das in $\mathcal{B}$ erfüllt ist, dann *stimmt t mit u überein* Wenn $\mathcal{B}$ konsistent ist, ist dies äquivalent zur Aussage: t stimmt mit u genau dann überein, wenn '$t \approx u$' in der transitiven Hülle von $\mathcal{B}$ ist.

Pläne. Ein *partiell geordneter, partiell instanziierter Plan* (kurz: ein Plan) ist ein Tripel $\mathcal{P} = (\mathcal{S}, \mathcal{O}, \mathcal{B})$, wobei $\mathcal{S}$ eine Menge von Schritten, $\mathcal{B}$ eine Menge von Bindungs-Constraints auf den Termen von $\mathcal{P}$ (genauer, die Terme in $\mathcal{S}$) und $\mathcal{O}$ eine Menge von Ordnungs-Constraints auf die Schritte in $\mathcal{S}$ ist.[4]

$\mathcal{P}$ ist *vollständig*, wenn $\mathcal{P}$ total geordnet und variablenfrei ist, d.h. es gibt eine eindeutige totale Ordnung $s_1 \prec s_2 \prec \ldots \prec s_n$ über $\mathcal{S}$, die $\mathcal{O}$ erfüllt, und eine eindeutige Grundsubstitution θ für die Terme von $\mathcal{P}$, die $\mathcal{B}$ erfüllen.[5] Ansonsten ist $\mathcal{P}$ *unvollständig.*

Eine *Vervollständigung* einer Menge von Ordnungs-Constraints $\mathcal{C}$ ist eine totale Ordnung $\mathcal{O}$, so daß gilt: $s\mathcal{C}t \rightarrow s\mathcal{O}t$ für s und t aus der selben Menge von Schritten. Jede Ordnung in einem unvollständigen Plan muß auch im vollständigen Plan gelten

Ein Plan $\mathcal{P}' = (\mathcal{S}', \mathcal{O}', \mathcal{B}')$ ist eine *Beschränkung* (constrainment) eines Planes $\mathcal{P} = (\mathcal{S}, \mathcal{O}, \mathcal{B})$, wenn $\mathcal{S}' = \mathcal{S}$, $\mathcal{O} \subseteq \mathcal{O}'$, $\mathcal{B} \subseteq \mathcal{B}'$. Eine *Vervollständigung* von $\mathcal{P}$ ist jede Beschränkung von $\mathcal{P}$, die vollständig ist.[6]

$\mathcal{P}$ ist *konsistent*, wenn $\mathcal{P}$ mindestens eine Vervollständigung hat; ansonsten ist $\mathcal{P}$ *inkonsistent* und nicht mehr ein gültiger unvollständiger Plan. An diesem Punkt setzt Backtracking an.

Im Sinne der Modallogik ist nun jeder Planzustand $(\mathcal{S}, \mathcal{O}, \mathcal{B})$, sowie insbesondere auch jede Vervollständigung eines Planes ein Knoten im Graphen der möglichen Welten. Die Planverfeinerungsoperationen, die einen Plan $(\mathcal{S}, \mathcal{O}, \mathcal{B})$ in einen Plan $(\mathcal{S}', \mathcal{O}', \mathcal{B}')$ überführen, definieren die Erreichbarkeits- oder Übergangsrelation (vgl. [Ric92], siehe dazu auch Abbildung 2.1).

Zustand in einer Planungswelt. Die Ausführung eines Planschrittes verändert die zugrundeliegende Planungswelt. Ein *Zustand* oder eine *Situation* in der Planungswelt ist eine endliche Menge von Grundatomen aus $\mathcal{L}$. Enthält ein Zustand z ein Grundatom p, dann ist p erfüllt in z und $\neg p$ nicht erfüllt; ansonsten ist p nicht erfüllt und $\neg p$ erfüllt.[7]

An dieser Stelle ist es wichtig, genau zwischen den beiden Begriffen *Zustand* und *Planzustand* (oder einfacher: Plan) zu unterscheiden, um Unklarheiten bzgl. der Zusammenhänge mit der

[4] Ein Startzustand, wie er von Kambhampati und Nau definiert wird, muß nicht explizit eingeführt werden, da er sich aus der Problembeschreibung ergibt, die mittels zweier ausgezeichnteter Schritte repräsentiert ist.

[5] Kurzschreibweise bei Kambhampati und Nau für einen vollständigen Plan $\mathcal{P} : s_1 \prec s_2 \prec \ldots \prec s_n$.

[6] Die Definition einer Vervollständigung in [Cha87] ist nicht eindeutig darin, ob nur die Schritte von $\mathcal{P}$ in der Vervollständigung enthalten sind oder ob noch andere Schritte hinzugefügt werden dürfen. Aus Chapmans weiteren Formulierungen geht jedoch hervor, daß er nur die Schritte von $\mathcal{P}$ meint.

[7] Daraus folgt, daß ein Zustand eine Herbrand-Interpretation für die Sprache $\mathcal{L}$ ist und jede Formel der Prädikatenlogik erster Stufe entweder in z gültig ist oder nicht, und zwar bzgl. der üblichen Definition von Gültigkeit in der Prädikatenlogik erster Stufe.

Modallogik zu vermeiden. Zustände beziehen sich immer auf einen bestimmten Plan und beschreiben die durch Planausführung hervorgerufenen Veränderungen, ausgehend von einer Startsituation in der Planungswelt. Die Gesamtheit der verschiedenen Pläne beschreibt die möglichen Welten im modallogischen Sinne.

Jeder Planschritt $s \in \mathcal{S}$ besitzt eine *Eingabesituation* $in(s)$, eine Menge von Aussagen, die vor der Ausführung dieses Schrittes wahr sind, und eine *Ausgabesituation* $out(s)$, eine Menge von Aussagen, die nach der Ausführung dieses Schrittes gelten. Ein Schritt *sichert eine Aussage in seiner Ausgabesituation zu*, wenn die Aussage mit einer Nachbedingung dieses Schrittes übereinstimmt. Die Ausgabesituation des Startschrittes s_0 heißt auch die *Startsituation* $init$, die Ausgabesituation des Zielschrittes heißt auch die *Zielsituation* fin. Die Startsituation liegt *vor*, die Zielsituation liegt *hinter* jeder anderen Situation. Alle diese Symbole müssen sich unterscheiden. Zu beachten ist, daß bei den Arbeiten [Cha87, KN95] die beiden Begriffe Planschritt und Ausgabesituation eines Planschrittes oft synonym verwendet werden. Gemeint ist mit 'in der Situation s gilt p', wobei s ein Planschritt ist, in allen Fällen 'in der Ausgabesituation von s gilt p'. Diese Sprechweise wird auch hier beibehalten. Dadurch ist es auch möglich, ausgehend von der Ordnung der Schritte eines Planes, eine Ordnung auf den von ihnen erzeugten Situationen zu definieren: Wenn $s, t \in \mathcal{S}$ und $s \prec t$ gilt, gilt auch $x \prec y$, wobei x gleich s, $in(s)$ oder $out(s)$, und y gleich t, $in(t)$ oder $out(t)$ sein kann.

Ausführbarkeit. Sei $\mathcal{P}$ ein vollständiger Plan und $k \leq n$ die größte ganze Zahl, für die Zustände $z_1, z_2, \ldots, z_k$ existieren, so daß für $1 \leq i \leq k$ folgende Eigenschaften gelten:

1. z_{i-1} erfüllt alle Vorbedingungen von s_i (d.h. $p\theta$ ist wahr in z_{i-1} für jedes $p \in pre(s_i)$).
2. z_i ist der Zustand, der durch die Ausführung von Schritt s_i im Zustand z_{i-1} entsteht, d.h. $z_i = (z_{i-1} - f_i) \cup t_i$, wobei t_i die Menge aller Grundatome $p\theta$ ist, $p \in post(s_i)$, und f_i die Menge aller negierten Grundatome $p\theta$ ist, $p \in post(s_i)$.

Dann gilt für $1 \leq i \leq k$: s_i ist im Eingabezustand z_{i-1} *ausführbar* mit dem Ausgabezustand z_i. Wenn $k = n$ ist, dann ist $\mathcal{P}$ ausführbar und liefert den Zielzustand z_n.

Wahrheit in einer Situation. Was ist nun *wahr* und was *falsch* in einer Situation eines vollständigen Plans?

Sei $\mathcal{P}$ ein vollständiger Plan und p ein Grundliteral. Dann ist p *wahr* in $init$, wenn p wahr ist im Startzustand von $\mathcal{P}$, und *wahr* in fin, wenn p wahr ist im Zielzustand von $\mathcal{P}$. Wenn s_i ein ausführbarer Schritt von $\mathcal{P}$ ist, dann ist p *wahr* in $in(s_i)$ (oder $out(s_i)$), wenn p wahr ist im Eingabezustand von s_i (bzw. im Ausgabezustand).

Ein Grundliteral p ist *falsch* in einer Situation z genau dann, wenn p nicht wahr ist in z (In der klassischen Logik ist dies gleichbedeutend mit: "wenn $\neg p$ wahr ist"). Chapman sagt dann, p wird in z *ungültig gemacht.* Wenn $\mathcal{P}$ nicht ausführbar ist, dann ist p weder wahr noch falsch in der Zielsituation von $\mathcal{P}$.

Aus dem genannten ergibt sich folgende Definition:

Notwendige (mögliche) Wahrheit: Es gilt *notwendigerweise* (*möglicherweise*) p genau dann, wenn p in *allen* (*einigen*) Vervollständigungen eines Planes wahr ist.

2.3 Probleme in Chapmans Definitionen

Chapman definiert *Situationen* als Mengen von Literalen. Wenn p ein Literal und s eine Situation ist, dann sagt Chapman: p ist wahr, wenn es mit einem Literal in s übereinstimmt; p ist falsch, wenn es mit der Negation eines Literals in s übereinstimmt.

Seine Definitionen von Situationen führen zu folgenden Problemen:

1. Bereits in [YT90] wurde gezeigt, daß ein Plan nicht vollständig ist, wenn seine Situationen nicht wohldefiniert sind.

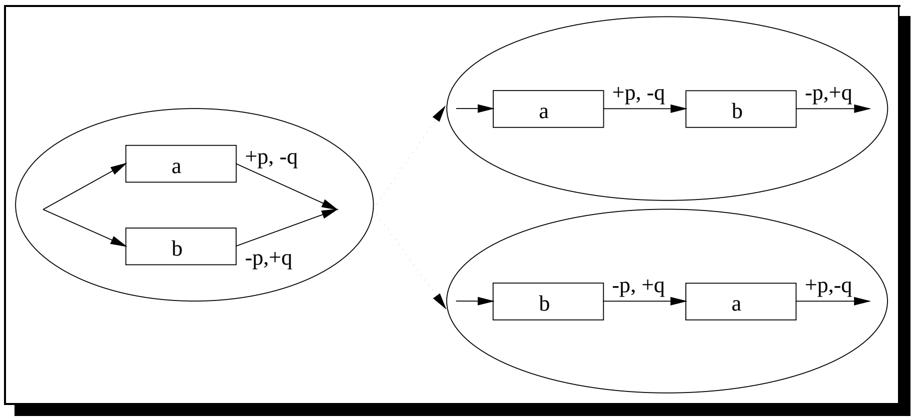


Abbildung 2.1: Zielsituation abhängig von Vervollständigung.

Beispiel: In Abbildung 2.1 ist links ein Plan $\mathcal{P}$ bestehend aus zwei ungeordneten Schritten a und b (Kästchen) dargestellt. p und q sind Prädikate. Es gilt: a sichert p zu und macht q ungültig, und b sichert q zu und macht p ungültig. Dann gilt in der Zielsituation von $\mathcal{P}$ entweder p oder q, abhängig von der gewählten Vervollständigung.

2. Wenn eine Situation ein Literal enthält, das noch Variablen beinhaltet (d.h. das Literal ist kein Grundliteral), dann ist die Bedeutung dieses Literals problematisch.

Beispiel: Die Startsituation eines Plans enthalte das Literal $p(x)$, wobei x eine Variable ist. Die Bedeutung kann nicht $\exists x : p(x)$ sein, da TWEAK später möglicherweise einen Constraint $x \neq y$ einführt für eine Konstante oder eine Variable y. Andererseits kann die Bedeutung auch nicht $\forall x : p(x)$ lauten, da eventuell ein Constraint $x = y$ erzeugt wird. Also ist $p(x)$ für irgendein x gemeint, das TWEAK irgendwann während des Planungsprozesses festlegt. Daraus folgt: TWEAK verändert die Bedeutung des Startzustandes während des Planens, wenn im Startzustand Variablen vorkommen.

Aus diesen beiden Gründen verwenden Kambhampati und Nau STRIPS-artige Zustände und definieren daraus folgend Situationen aus Zuständen. Ist ein Plan nach ihrem Verständnis vollständig und kann er mindestens bis zur Situation z ausgeführt werden, so korrespondiert z mit einer Situation t, die während der Planausführung entsteht. Und all das, was in z wahr bzw. falsch ist, ist wahr bzw. falsch in t.[8] Ansonsten ist *nichts* wahr oder falsch in z, wenn es auch bestimmte Aussagen gibt, die bedingt wahr oder falsch sind (vgl. Kapitel 4).

[8] Bei Chapman ist der Wahrheitswert einer Aussage p erst dann in einer Situation z definiert, wenn p oder $\neg p$ mit irgendeiner Aussage in z übereinstimmt. Kamhampati und Nau können jederzeit angeben, ob p in einer Situation z gültig ist, oder nicht.

Kapitel 3

Chapmans Modal Truth Criterion

Das nichtlineare Planungssystem TWEAK wurde von David Chapman entwickelt, um den formalisierten Planungsansatz zu implementieren, den er in [Cha87] vorstellt und der im folgenden erläutert werden soll, da er der Ausgangspunkt für spätere Kapitel ist. Grundlegendes Element ist Chapmans *modales Wahrheitskriterium* (Modal Truth Criterion), welches feststellt, ob eine Aussage in einer bestimmten Situation notwendigerweise wahr ist.

3.1 Das Modal Truth Criterion

Betrachten wir noch einmal das am Ende von Abschnitt 2.2 auf Seite 6 genannte Kriterium für die Bestimmung der notwendigen bzw. möglichen Wahrheit einer Aussage für einen bestimmten Plan $\mathcal{P}$, so bezieht sich das dortige Kriterium in direkter Anlehnung an die Definitionen aus der Modallogik auf einige oder alle Vervollständigungen des Plans, die modallogisch erreichbare Welten des Plans sind. Für ein algorithmisches Abprüfen des Kriteriums ist diese Definition jedoch wegen der Notwendigkeit, einige oder alle Vervollständigungen eines Planes zu berechnen, ungeeignet.

David Chapmans Leistung war es, den modalen Wahrheitsgehalt einer Aussage p für einen Plan $\mathcal{P}$ in einer Weise abprüfbar zu machen, die auf die Berechnung solcher Vervollständigungen verzichten kann. Diese Form des modalen Wahrheitskriteriums ist unter dem Namen *Modal Truth Criterion (MTC)* bekannt. Es definiert, basierend auf einem Plan $\mathcal{P}$, ein Kriterium dafür, wann eine Aussage p notwendigerweise bzw. möglicherweise wahr ist.

Die Bedeutung eines Planes ist während des Planungsvorgangs unsicher. Dies ist bedingt durch dessen Unvollständigkeit, dessen partielle Ordnung. In der Klötzchenwelt kann man sich dies folgendermaßen vorstellen: Wenn x eine Variable ist, kann man nach der Zusicherung $On(block, x)$ nicht sagen, ob $On(block, c)$ wahr oder falsch ist, solange x nicht notwendigerweise mit einer bestimmten Konstanten übereinstimmt, x also gebunden ist.

Notwendige Wahrheit in Situationen. Offensichtlich ist eine Aussage notwendigerweise wahr in einer Situation, wenn sie in ihr zugesichert wurde. Sobald eine Aussage zugesichert wurde, bleibt sie solange wahr, bis sie durch einen Schritt ungültig gemacht wird. Demnach ist eine Aussage p notwendigerweise wahr in einer Situation, wenn p in einer vorhergehenden Situation notwendigerweise wahr ist und es keinen Schritt gibt, der p ungültig machen könnte. Denn, wenn ein Zwischenschritt existiert, der p ungültig machen könnte (z.B. indem er ein q ungültig macht, das mittels geeigneter Constraints mit p übereinstimmt), dann gibt es eine

Vervollständigung, in der der Zwischenschritt tatsächlich dazwischen liegt und p ungültig macht.

Andererseits zeigt Abbildung 3.1 einen unvollständigen Plan, bestehend aus drei Schritten (Kästchen) und den Prädikaten p und q. Die Pfeile geben die zeitliche Ordnung wieder, in der

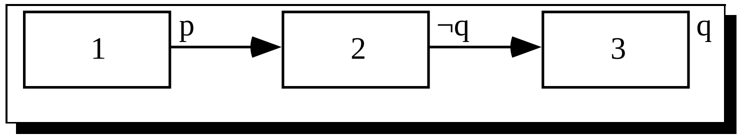


Abbildung 3.1: Ein unvollständiger Plan.

die Schritte stehen. Planschritt 1 sichert p zu, Planschritt 2 sichert $\neg q$ zu und macht somit q ungültig, Planschritt 3 sichert q zu. Der Plan kann nun auf zwei Arten vervollständigt werden: Zum einen, indem p und q tatsächlich übereinstimmen, so daß p durch Schritt 3 zugesichert wird, zum anderen, indem p und q nicht übereinstimmen, so daß p durch Schritt 1 zugesichert und nicht mehr ungültig gemacht wird. In beiden Fällen ist p in der Zielsituation wahr, obwohl kein Schritt p notwendigerweise zusichert und kein Schritt dazwischenliegt, der p ungültig machen könnte.

Hier nun das vollständige Kriterium, das auch den letztgenannten Fall berücksichtigt. Im Gegensatz zur Definition aus Abschnitt 2.2, die zur Prüfung der notwendigen Wahrheit einer Aussage die Berechnung aller Vervollständigungen verlangt, gibt Chapman hier Bedingungen an, die den Test anhand eines einzigen Planes erlauben.

Modal Truth Criterion. Gegeben sein ein Plan $\mathcal{P} = (\mathcal{S}, \mathcal{O}, \mathcal{B})$. Eine Aussage p ist in der Ausgabesituation eines Schrittes $s \in \mathcal{S}$ genau dann *notwendigerweise wahr*, wenn folgende Bedingungen erfüllt sind:

1. Es existiert ein Schritt $t \in \mathcal{S}$, der entweder gleich s ist oder bzgl. der Ordnung $\mathcal{O}$ vor s liegt und p zusichert.
2. Für jeden Schritt $C \in \mathcal{S}$, der bzgl. $\mathcal{O}$ vor s liegt und alle Aussagen q, die mit p übereinstimmen können und die C ungültig macht, existiert ein Schritt $W \in \mathcal{S}$ zwischen C und s (bzgl. $\mathcal{O}$), der eine Aussage r zusichert, mit der Eigenschaft: r und q stimmen immer dann überein, wenn p und q übereinstimmen.

Das Kriterium soll nun anhand von Abbildung 3.2 näher erläutert werden. Durch normale

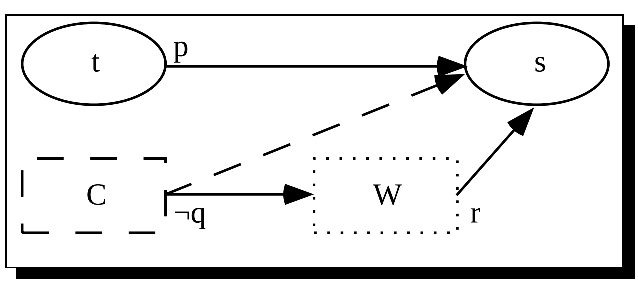


Abbildung 3.2: Das MTC für notwendige Wahrheit

Linien verbundene Schritte stehen in notwendiger zeitlicher Ordnung zueinander. Die beiden durch eine gestrichelte Linie verbundenen Schritte stehen in einem möglichen zeitlichen

Zusammenhang. Das Kästchen mit gestrichelter Umrandung (C) zeigt einen unerlaubten Schritt. Das gepunktete Kästchen (W) stellt einen Schritt dar, der den gestrichelten Schritt zu einem erlaubten machen würde.

Chapman nennt einen Schritt t, der bzgl. $\mathcal{O}$ vor s liegt und p zusichert, *Establisher* (etwa: Erzeuger). Ein Schritt C, wie oben definiert, heißt *Clobberer* (etwa: Zerstörer), und ein Schritt W, der die Auswirkungen von Schritten C verhindert, die ansonsten Clobberer werden würden, nennt er *White Knight* (weißer Ritter).

Ist ein Schritt C vor t angeordnet, dann kann C nicht zum Zerstörer für t bzgl. p werden. In solch einem Fall ist der Schritt t selbst der White Knight. Der Teil des MTC, der vom White Knight handelt, ist nicht unmittelbar einsichtig, wird aber benötigt, wie der Plan aus Abbildung 3.1 zeigte.

3.2 Prozedurale Interpretation des MTC: TWEAK

Im nichtlinearen Planungssystem TWEAK verwendete Chapman das MTC zur Planung.[1] TWEAK ist ein sogenannter Constraint-propagierender Planer. Constraint-Propagierung heißt, es wird ein Plan erzeugt, indem das System nach und nach partielle Beschreibungen (Constraints, einschränkende Bedingungen) spezifiziert, die der Plan erfüllen muß. Man kann die Constraint-Propagierung auch als Suchstrategie auffassen. Stück für Stück werden dabei Teile des Suchraums durch Betrachtung von Constraints entfernt.

Der Vorteil des Constraint-propagierenden Ansatzes liegt in der Tatsache, daß erst dann Eigenschaften eines gesuchten Objektes festgelegt werden müssen, wenn eine begründete Entscheidung gefällt werden kann. Oft wird durch diese Reduzierung zufälliger Wahlmöglichkeiten die Häufigkeit von Backtracking vermindert.

TWEAK hat zu jedem Zeitpunkt einen unvollständigen Plan, eine partielle Beschreibung eines Planes, der möglicherweise die gestellte Aufgabe löst. Der unvollständige Plan enthält partielles Wissen des vollständigen Plans, der schließlich ausgewählt wird. Der Planungsvorgang ist abgeschlossen, wenn alle Vervollständigungen des unvollständigen Plans das gegebene Problem lösen.

Das MTC ist auch prozedural interpretierbar. Chapmans *Goal-Achievement-Prozedur* interpretiert das Wahrheitskriterium als nicht-deterministischen Algorithmus, wie in Abbildung 3.3 dargestellt. Die All-Quantifizierung über eine Menge wird dabei zur Iteration über deren Elemente. Der Existenz-Quantor wird als nicht-deterministische Auswahl interpretiert. Ebenfalls nicht-deterministisch ist die Disjunktion zu verstehen, während Konjunktionen die Ausführung aller Pfade verlangt. Eine existentiell quantifizierte Situation kann nicht-deterministisch erfüllt werden, indem entweder ein bereits bestehender Schritt verwendet (Simple Establishment) oder ein neuer Schritt zum Plan hinzugefügt (Step Addition) wird. Der neu hinzugefügte Schritt muß eine Aktion sein, die in der gew"weißen Ritterahlten Domäne ausführbar sein muß. Die Wahl besteht aus den Schritten, die in der Domäne erlaubt sind und das gewünschte Ziel möglicherweise zusichern.

Der Algorithmus fügt Constraints in den Plan ein, um zwei Aussagen zu ordnen (Promotion) oder übereinstimmen zu lassen (Separation bzw. White Knight Declobbering). Für die Konsistenz der Constraints ist ein Constraint-Verwaltungsmodul zuständig, das ggf. Backtracking anstößt.

[1] Eine algorithmische Version ist z.B. in [KY95] finden.

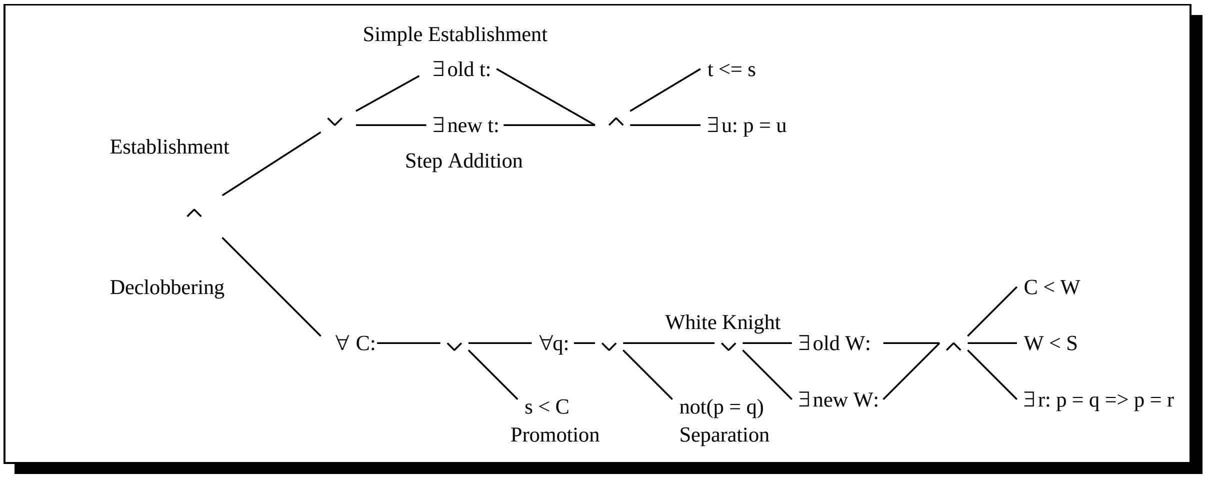


Abbildung 3.3: Die nichtdeterministische Goal-Achievement-Prozedur von Chapman

Da das Wahrheitskriterium bzgl. notwendiger Wahrheit sowohl hinreichend als auch notwendig ist, durchläuft die Goal-Achievement-Prozedur alle Wege, die zum Ziel führen. Der Algorithmus benötigt polynomiale Zeit bzgl. der Anzahl von Planschritten. Der Rumpf des MTC kann für jedes der n^3 Tripel $< t, C, W >$ mit einer festen Menge von Aufrufen eines in polynomialer Zeit arbeitenden Constraint-Verwaltungsmoduls verifiziert werden.

Ein Beweis für die Korrektheit des MTC bzgl. notwendiger Wahrheit, findet sich im Anhang von [Cha87].

3.3 Probleme

Chapman behauptet, daß in seiner Definition des MTC ein einfaches Ersetzen von *notwendigerweise* durch *möglicherweise* und umgekehrt ausreicht, um *möglicherweise wahr* zu testen. Ein erstes Problem taucht hierbei bereits durch das Vermischen der umgangssprachlichen mit der von der Modallogik definierten Bedeutung von 'möglicherweise' und 'notwendigerweise' auf. Chapmans Behauptung ist außerdem falsch und wurde von Subbarao Kambhampati und Dana S. Nau widerlegt. Die Begründung und eine Erweiterung stellt das anschließende Kapitel vor.

Kambhampati und Nau widmeten sich in [KN94, KN95] der wesentlichen Idee Chapmans, der Idee des MTC. Sie weisen nach, daß dessen Definition für mögliche Wahrheit nicht korrekt ist. Chapman stellt nur notwendige, aber nicht hinreichende Bedingungen zur Verfügung.

Außerdem zeigen sie, daß sich nur notwendige Wahrheit in polynomialer Zeit berechnen läßt. Für mögliche Wahrheit gilt dies nicht, da dieses Problem NP-schwierig und somit nur mit exponentiellem Aufwand berechenbar ist.

Kapitel 4

Das Modal Conditional Truth Criterion

Subbarao Kambhampati und Dana S. Nau formulieren ein Konzept, das sie *modale bedingte Wahrheit* (modal conditional truth) nennen, das aber, anders als Chapmans modale Wahrheit, die Ausführbarkeit eines Planes nicht fordert. Hierbei sind notwendige bedingte Wahrheit und mögliche bedingte Wahrheit tatsächlich Komplemente zueinander und beide können in polynomialer Zeit berechnet werden. Wird das MTC mit Hilfe modaler bedingter Wahrheit definiert, dann ist das MTC sowohl für notwendige bedingte Wahrheit als auch für mögliche bedingte Wahrheit korrekt, und beide sind dual zueinander.[1]

Im ersten Abschnitt wird der Begriff der bedingten Wahrheit eingeführt. Desweiteren werden in diesem Zusammenhang verschiedene Entscheidungsprobleme bzgl. modaler und modaler bedingter Wahrheit formuliert.
Darauf aufbauend wird gezeigt, daß die Berechnung notwendiger Wahrheit zwar in polynomialer Zeit erfolgen kann, dies aber für mögliche Wahrheit nicht gilt, da die beiden Kriterien nicht dual zueinander sind.
Der dritte Abschnitt widmet sich einem Vergleich mit Chapmans MTC, betrachtet modale Dualität und universelle Ausführbarkeit und stellt schließlich verschiedene Forschungsansätze vor, die sich mit universeller Ausführbarkeit beschäftigen.

4.1 Wahrheit in Plänen

4.1.1 Wahrheit und bedingte Wahrheit

Aus den Definitionen für Wahrheit in einer Situation (vgl. Abschnitt 2.2) folgt, daß p in s wahr ist (kurz: $M(p,s)$), wenn folgende drei Bedingungen erfüllt sind:

Establishment (Erzeugung): Entweder stimmt p mit einer Nachbedingung eines Schrittes a überein, so daß $a \prec s$, oder $p \in s_0$.

Nondeletion (Nicht-Zerstörung): Für alle Schritte b zwischen a (oder s_0) und s stimmt keine Nachbedingung von b mit $\neg p$ überein (Dies ist im Prinzip der Declobbering-Teil von Chapmans MTC).

[1] Sowohl Chapman als auch Kambhampati und Nau gehen desweiteren auf die Rolle des MTC in der Plangenerierung ein, siehe dazu [Cha87, KN95, McD91].

Executability (Ausführbarkeit): Jeder Schritt, der vor s liegt, ist ausführbar.

Ein ähnliches Konzept ist Kambhampati und Naus *bedingte Wahrheit*, die sich von der oben definierten Wahrheit nur im Fehlen der Ausführbarkeit unterscheidet. Es gilt: p ist *bedingt wahr* in s (kurz $C(p,s)$) genau dann, wenn die beiden Bedingungen Establishment und Non-deletion erfüllt sind.

4.1.2 Modale Wahrheit

Wahrheit und bedingte Wahrheit sind nur für vollständige Pläne definiert, da es bei unvollständigen Plänen von der gewählten Vervollständigung abhängt, was wahr oder bedingt wahr ist. Deshalb muß bei unvollständigen Plänen modale Wahrheit verwendet werden.

Zur Erinnerung noch einmal die Definition Chapmans für modale Wahrheit:

- Es gilt *notwendigerweise* p genau dann, wenn p in *allen* Vervollständigungen eines unvollständigen Plans wahr ist.
- Es gilt *möglicherweise* p genau dann, wenn p in *einigen* Vervollständigungen eines unvollständigen Plans wahr ist.

In dieser Definition meint Chapman offenbar irgend eine beliebige Behauptung über einen Plan. Seine Beispiele enthalten nicht nur Aussagen über bestimmte Literale und Situationen, sondern auch Behauptungen über den ganzen Plan (z.B., daß ein Plan ein Problem notwendigerweise löst). Für formale Resultate ihres Papiers beschränken Kambhampati und Nau deshalb die Natur von p.

Sei p ein Grundliteral, $\mathcal{P}$ ein Plan und s eine Situation in $\mathcal{P}$. Dann gilt:

- p ist *notwendigerweise* (oder *möglicherweise*) wahr in s ($\Box M(p,s)$ bzw. $\Diamond M(p,s)$) genau dann, wenn $M(p,s)$ in allen (manchen) Vervollständigungen von $\mathcal{P}$ gilt.
- p ist notwendigerweise (oder möglicherweise) *bedingt* wahr in s ($\Box C(p,s)$ bzw. $\Diamond C(p,s)$) genau dann, wenn $C(p,s)$ in allen (manchen) Vervollständigungen von $\mathcal{P}$ gilt.

Weiterhin definieren Kambhampati und Nau in diesem Zusammenhang folgende Entscheidungsprobleme:

Notwendige Wahrheit: Ist p *notwendigerweise* wahr im Zielzustand von $\mathcal{P}$? (Erzeugt jede Vervollständigung von $\mathcal{P}$ einen Zielzustand, in dem p wahr ist?)

Mögliche Wahrheit: Ist p *möglicherweise* wahr im Zielzustand von $\mathcal{P}$? (Gibt es eine Vervollständigung von $\mathcal{P}$, die einen Zielzustand erzeugt, in dem p wahr ist?)

Notwendige bedingte Wahrheit: Ist p *notwendigerweise bedingt* wahr im Zielzustand von $\mathcal{P}$?

Mögliche bedingte Wahrheit: Sei p ein Grundliteral und $\mathcal{P}$ ein Plan. Ist p *möglicherweise bedingt* wahr im Zielzustand von $\mathcal{P}$?

4.2 Dualität und Komplexität

Dieser Abschnitt stellt zum einen die Resultate [KN94, KN95] vor, die bzgl. der Nicht-Dualität von notwendiger und möglicher Wahrheit gefunden wurden, und zum anderen die Komplexität der Berechnung möglicher Wahrheit.

4.2.1 Modale Dualität bzw. Nichtdualität

Berechnung notwendiger Wahrheit. Aus den Definitionen für Wahrheit und bedingte Wahrheit im vorhergehenden Abschnitt ist leicht ersichtlich, daß p in jeder Vervollständigung eines Planes $\mathcal{P}$ im Zielzustand fin genau dann wahr ist, wenn

1. p in fin bedingt wahr und
2. die Vervollständigung selbst ausführbar ist.

Eine Vervollständigung ist genau dann *ausführbar*, wenn für jede Aktion a und jede Vorbedingung p_a von a gilt: p_a ist bedingt wahr in der Situation $in(a)$.

Daraus ergibt sich folgende Gleichung:

$$\mathcal{M}(p, fin) \equiv \left(\mathcal{C}(p, fin) \wedge \underbrace{\left(\bigwedge_{\forall a \in \mathcal{P}, \forall p_a \in pre(a)} \mathcal{C}(p_a, in(a)) \right)}_{Ausf\ddot{u}hrbarkeit\ dieser\ Vervollst\ddot{a}ndigung} \right) \tag{4.1}$$

Da die Äquivalenz von p *ist notwendigerweise wahr im Zielzustand von* $\mathcal{P}$ zu p *ist wahr im Zielzustand jeder Vervollständigung von* $\mathcal{P}$ gegeben ist, gilt:

$$\Box\mathcal{M}(p, fin) \equiv \Box \left(\mathcal{C}(p, fin) \wedge \left(\bigwedge_{\forall a \in \mathcal{P}, \forall p_a \in pre(a)} \mathcal{C}(p_a, in(a)) \right) \right) \tag{4.2}$$

Aus der Tatsache, daß das Distributivgesetz für notwendige modale Wahrheit bzgl. des Verknüpfungsoperators $\wedge$ gilt $(\Box(p \wedge q) \equiv (\Box p \wedge \Box q))$, folgt:

$$\Box\mathcal{M}(p, fin) \equiv \left(\Box\mathcal{C}(p, fin) \wedge \underbrace{\left(\bigwedge_{\forall a \in \mathcal{P}, \forall p_a \in pre(a)} \Box\mathcal{C}(p_a, in(a)) \right)}_{Ausf\ddot{u}hrbarkeit\ aller\ Vervollst\ddot{a}ndigungen} \right) \tag{4.3}$$

Ob p nun notwendigerweise wahr ist, läßt sich also berechnen, indem man einerseits prüft, ob p notwendigerweise bedingt wahr ist, und andererseits alle Vorbedingungen jedes Schrittes daraufhin untersucht, ob sie notwendigerweise bedingt wahr sind. Dies läßt sich mittels eines Constraint-Verwaltungsmoduls durchführen, wie Chapman bereits vorschlug. Die Berechnung der notwendigen bedingten Wahrheit eines Literals in einer Situation kann in polynomialer Zeit ($O(n^3)$) bzgl. der Planlänge geschehen, wobei die Planlänge gleich der Anzahl der Schritte ist.[2]

Berechnung möglicher Wahrheit. Für den Fall der möglichen Wahrheit ergibt sich auf ähnliche Weise die folgende Gleichung:

$$\Diamond\mathcal{M}(p, fin) \equiv \Diamond\left(\mathcal{C}(p, fin) \wedge \left(\bigwedge_{\forall a\in\mathcal{P}, \forall p_a\in pre(a)} \mathcal{C}(p_a, in(a))\right)\right) \tag{4.4}$$

Aber für mögliche Wahrheit gilt das Distributivgesetz **nicht** ($\Diamond(p \wedge q) \not\equiv (\Diamond p \wedge \Diamond q)$. Gleichung 4.4 läßt sich also nicht vereinfachen, woraus folgt, daß notwendige Wahrheit in fin und mögliche Wahrheit in fin *nicht* dual zueinander sind, wenn dies auch für notwendige und mögliche bedingte Wahrheit gilt.

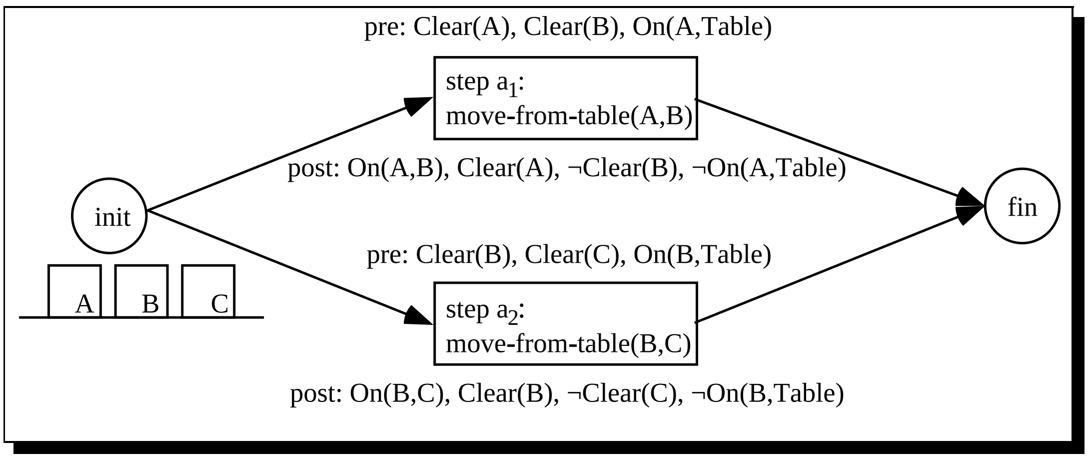


Abbildung 4.1: Nichtdualität notwendiger und möglicher Wahrheit in TWEAK-Plänen.

Satz 1: Es gibt ein Grundliteral p und einen Plan $\mathcal{P}$, so daß im Zielzustand fin von $\mathcal{P}$ gilt: $\Box\mathcal{M}(p, fin) \not\equiv \neg\Diamond\neg\mathcal{M}(p, fin)$.

Beweis: Die Korrektheit dieser Aussage läßt sich leicht mit Hilfe von Abbildung 4.1 beweisen. Sie zeigt einen unvollständigen Plan aus der Blockwelt in dem folgendes in der Startsituation *init* gilt: $On(A, Table)$, $On(B, Table)$, $On(C, Table)$,$Clear(A)$, $Clear(B)$, $Clear(C)$. Der Operator *move-from-table*(x, y) nimmt x vom Tisch und legt ihn auf y. Über jedem Schritt stehen die entsprechenden Vorbedingungen, darunter die Nachbedingungen. Der Plan besitzt nur eine einzige ausführbare Vervollständigung: $a_2 \prec a_1$. Diese Vervollständigung erzeugt einen Zielzustand, in dem A auf B und B auf C liegt. Da keine ausführbare Vervollständigung erzeugt wird, in der $\neg On(A, B)$ wahr ist, ist $\neg On(A, B)$ nicht möglicherweise wahr im Zielzustand fin. Wenn mögliche und notwendige Wahrheit dual zueinander wären, würde dies bedeuten, daß $On(A, B)$ notwendigerweise wahr in fin wäre. Es gilt jedoch, daß $On(A, B)$ nicht notwendigerweise wahr ist in fin, denn der Plan enthält eine nicht ausführbare Vervollständigung: $a_1 \prec a_2$. Daraus folgt, daß mögliche und notwendige Wahrheit nicht dual zueinander sind. $\Box$

[2]Da die Gesamtanzahl der Vorbedingungen in einem Plan in der gleichen Größenordnung liegt wie die Anzahl der Aktionen, kann also die Berechnung in polynomialer Zeit erfolgen.

Grund für Nicht-Dualität. Notwendige und mögliche modale Wahrheit definieren aufgrund fehlender Dualität keine wohlgeformte modale Logik (vgl.[Ric92]). Die Ursache dafür liegt darin, daß die Semantik einer Modallogik auf sogenannten Kripke-Strukturen (möglichen Welten) aufbaut. In dieser Formulierung gilt für ein Grundliteral p, daß es in jeder möglichen Welt entweder wahr oder falsch ist. Für partiell geordnete Pläne könnte man sagen, jede Vervollständigung stelle eine mögliche Welt dar. Die modale Wahrheit eines Prädikats p in einer Situation eines Planes erfordert jedoch die *Ausführbarkeit* der Aktionen, um zu eben dieser Situation zu gelangen. Die Wahrheit von p in der entsprechenden Welt ist aber nicht definiert, wenn die Ausführbarkeit nicht gewährleistet ist.

4.2.2 Komplexität notwendiger und möglicher Wahrheit

Wenn ein Grundliteral p und ein Plan $\mathcal{P}$ gegeben sind, ist p möglicherweise wahr im Zielzustand von $\mathcal{P}$ genau dann, wenn es eine ausführbare Vervollständigung von $\mathcal{P}$ gibt, die einen Zielzustand, in dem p wahr ist, erzeugt. Und dies ist genau dann der Fall, wenn nicht jede ausführbare Vervollständigung von $\mathcal{P}$ einen Zielzustand produziert, in dem $\neg p$ wahr ist.

Somit ist mögliche Wahrheit dual zu partieller Wahrheit, die von Kambhampati und Nau folgendermaßen definiert wird:

Partielle Wahrheit: Gegeben sei ein Grundliteral p und ein Plan $\mathcal{P}$. Erzeugt jede *ausführbare* Vervollständigung von $\mathcal{P}$ einen Zielzustand, in dem p wahr ist?

Der Beweis für die NP-Vollständigkeit partieller Wahrheit findet sich im Anhang von [KN95].

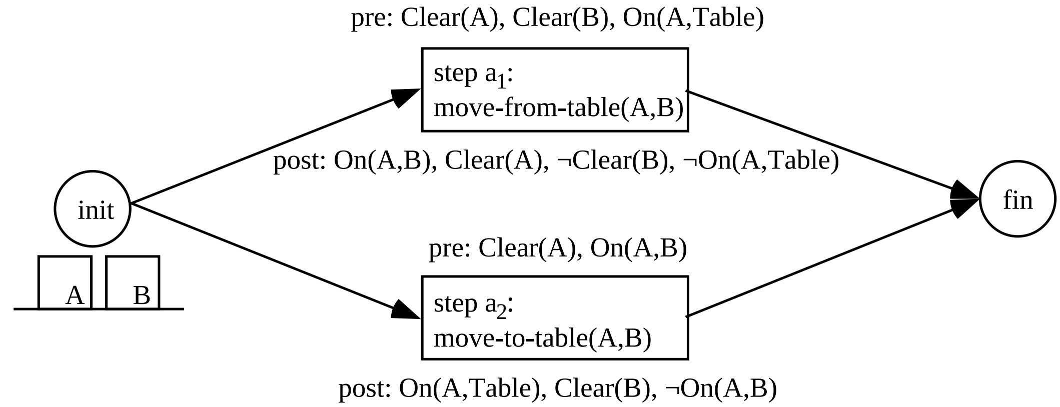
Partielle Wahrheit ist eine schwächere Bedingung als notwendige oder notwendige bedingte Wahrheit. Das Beispiel in Abbildung 4.2 illustriert dies. Alle ausführbaren Vervollständigungen erzeugen in manchen Fällen einen Zielzustand, in dem p wahr ist. Aber p ist weder notwendigerweise noch notwendigerweise bedingt wahr im Zielzustand.

Kambhampati und Nau zeigen noch einen anderen Weg, das Problem der Vereinfachung von Gleichung 4.4 zu verstehen. Wenn p möglicherweise bedingt wahr ist und alle Vorbedingungen der vorhergehenden Schritte möglicherweise bedingt wahr sind, impliziert dies nur, daß jede einzelne Vorbedingung für sich genommen in mindestens einer Vervollständigung wahr ist. Diese Bedingung ist notwendig, aber nicht hinreichend, um mögliche Wahrheit sicherzustellen.

Man könnte nun mögliche Wahrheit testen, indem man prüft, ob alle Bedingungen in einer Vervollständigung wahr sind. Da jedoch die Anzahl der Vervollständigungen eines Planes exponentiell bezüglich der Anzahl der Schritte des Planes wächst, würde dieser Test exponentielle Zeit beanspruchen. Dies besagt auch der folgende Satz aus [KN95].

Satz 2: Mögliche Wahrheit ist NP-vollständig.

Kambhampati und Nau zeigen hiermit, daß sich notwendige und mögliche Wahrheit auf unterschiedlichen Komplexitätsniveaus bewegen. Würde die modale Dualität gelten, wären folglich beide in polynomialer Zeit berechenbar.



Eigenschaft	**Atome mit dieser Eigenschaft**
Notwendigerweise wahr (von allen Vervollständigungen erzeugt)	*keines*
Bedingt notwendigerweise wahr (von allen Vervollständigungen erzeugt, wenn die Vorbedingungen weggelassen werden)	$Clear(A)$, $On(B, Table)$
Partiell wahr (von allen ausführbaren Vervollständigungen erzeugt)	$Clear(A)$, $On(A, Table)$ $Clear(B)$, $On(B, Table)$

Abbildung 4.2: Ein Beispiel dafür, daß partielle Wahrheit schwächer ist als notwendige und notwendige bedingte Wahrheit. Die Tabelle zeigt Atome, die die verschiedenen Wahrheitsbedingungen in fin erfüllen.

4.3 Vergleich mit Chapmans MTC

Die Analyse von Chapmans Arbeit durch Kambhampati und Nau ergab, daß die Definition des MTC bzgl. möglicher Wahrheit eines Prädikates fehlerhaft ist. Sie schlagen deshalb die Verwendung bedingter Wahrheit vor, durch die das Problem behoben werden kann.

Der zweite Unterabschnitt widmet sich universeller Ausführbarkeit zur Wiederherstellung modaler Dualität.

Im letzten Unterabschnitt schließlich werden verschiedene Forschungsansätze vorgestellt, die alle auf dem Prinzip der universellen Ausführbarkeit von Schritten beruhen.

4.3.1 Bedingte Wahrheit in Chapmans MTC

In Abschnitt 3.1 wurde Chapmans Definition des MTC erläutert. Nimmt man diese Definition wörtlich, dann besagt die Definition modaler Wahrheit, daß ein Plan modal ausführbar sein muß. Dies ist konsistent mit Chapmans Definition einer Situation (siehe Abschnitt 2.2), die aussagt, daß eine Ausgabesituation erst dann definiert ist, wenn der zugehörige Schritt ausgeführt werden konnte.

Kambhampati und Nau betrachteten den Beweis über notwendige und ausreichende Bedingungen für Chapmans MTC näher und stellten fest, daß Chapman mit notwendiger *bedingter*

Wahrheit anstelle von notwendiger Wahrheit arbeitete.

Durch den Beweis, daß jedes Literal mit einem Erzeuger und keinem Zerstörer notwendigerweise wahr ist, bezieht sich Chapman auf White-Knight-Schritte für jeden potentiellen Clobberer, ohne jedoch zu prüfen, ob die White-Knight-Schritte überhaupt ausführbar sind.

Dies hat freilich keine Konsequenzen für die notwendige Wahrheit, denn die Ausführbarkeit ist eine logische Folge der rekursiven Anwendung von notwendiger bedingter Wahrheit auf die vorhergehenden Schritte. Für die mögliche Wahrheit kann dies jedoch nicht garantiert werden, da, wie bereits erwähnt, für modale mögliche Wahrheit das Distributivgesetz nicht gilt. Daraus folgt, daß Chapmans Beweis nicht auf mögliche Wahrheit ausgedehnt werden kann.

Dies faßt folgender Satz von Kambhampati und Nau nochmal zusammen:

Satz 3: Es existiert ein Plan $\mathcal{P}$ und ein Grundliteral p, so daß gilt: In der Zielsituation von $\mathcal{P}$ ist p nicht möglicherweise wahr, aber das MTC behauptet das Gegenteil.

Beweis: Betrachtet werden soll noch einmal das Beispiel aus Abbildung 4.2. Die Bedingung $On(A, B)$ ist wahr in $out(a_1)$. a_2 löscht $On(A, B)$, aber a_1 erzeugt $On(A, B)$. Außerdem kann a_1 hinter a_2 folgen. Daraus schließt das MTC, daß $On(A, B)$ möglicherweise wahr ist in fin. Das ist jedoch falsch, da keine ausführbare Vervollständigung des Planes existiert, in dessen Zielzustand $On(A, B)$ wahr ist. □

Die obige Diskussion bewog Kambhampati und Nau zu einer alternativen Interpretation des MTC, um das genannte Problem zu umgehen. Sie lassen die Bedingung der Ausführbarkeit fallen und empfehlen, das MTC als eine Aussage über modale *bedingte* Wahrheit denn als Aussage über modale Wahrheit zu betrachten. Sie halten diese Interpretation für naheliegend, da Chapman die Wahrheit eines Literals ohne ausdrückliche Ausführbarkeit definierte, woraus sich ableiten läßt, daß das MTC nicht auf die Ausführbarkeit von $\mathcal{P}$ angewiesen ist.

Daraus wiederum schließen sie, daß Chapman eigentlich von modaler bedingter Wahrheit spricht (und vermutlich selbst bemerkt hätte, daß sein Beweis bei möglicher Wahrheit fehlerhaft ist, wenn er expilizit modale bedingte Wahrheit genannt hätte).

4.3.2 Modale Dualität und universelle Ausführbarkeit

Die fehlende Dualität notwendiger und möglicher Wahrheit wurde auf verschiedene Arten und Weisen angegangen. Kamhhampati und Nau stellen verschiedene Ansätze vor, die im folgenden Abschnitt wiedergegeben werden.

Ein Weg, Dualität zu erreichen, besteht darin, Pläne durchweg ausführbar zu halten. Ein Weg, die Ausführbarkeit von Plänen zu garantieren, besteht wiederum darin, nur Schritte ohne Vorbedingungen zuzulassen, d.h. nur Pläne zu gestatten, für die gilt: $pre(a) = \emptyset$ für jeden Schritt a von $\mathcal{P}$. In diesem Fall werden die Gleichungen 4.2 und 4.4 vereinfacht zu: $\Box\mathcal{M}(p, s) \equiv \Box\mathcal{C}(p, s)$ und $\Diamond\mathcal{M}(p, s) \equiv \Diamond\mathcal{C}(p, s)$. Mit anderen Worten: Modale und modale bedingte Wahrheit werden identisch in Plänen, deren Schritte ohne Vorbedingungen sind. Notwendige und mögliche Wahrheit werden dual zueinander und beide in polynomialer Zeit berechenbar.

Offensichtlich ist dieser Ansatz reichlich restriktiv. Eine Abschwächung könnte folgendermaßen aussehen. Eine Aktion a sei auch dann ausführbar, wenn ihre Vorbedingungen nicht erfüllt sind. Sind ihre Vorbedingungen erfüllt, dann wird a die entsprechenden Nachbedingungen erzeugen; wenn nicht, hat a keine Effekte.

In Plänen, die nur diese Art von Schritten beinhalten, sind mögliche und notwendige Wahrheit dual zueinander. Die Berechnung von möglicher Wahrheit ist NP-schwierig, die Berechnung notwendiger Wahrheit co-NP-schwierig. Abbildung 4.3 faßt die verschiedenen Komplexitätsrelationen der unterschiedlichen Entscheidungsprobleme zusammen.

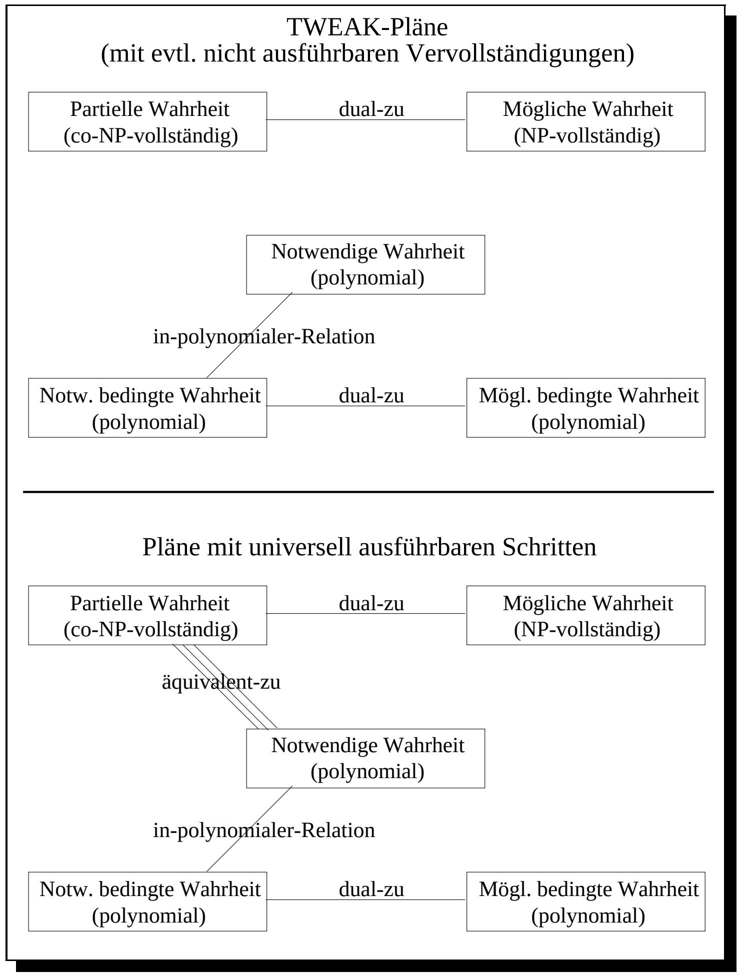


Abbildung 4.3: Komplexitätsrelationen zwischen Entscheidungsproblemen für Pläne.

4.3.3 Universelle Ausführbarkeit in anderen Ansätzen

Dynamische Aussagenlogik. Die dynamische Aussagenlogik ist eine Variante der Modallogik, die ursprünglich entwickelt wurde, um eine Semantik für Computerprogramme zu haben [Pra]. Der obige Ansatz wurde zuerst in [Ros81] verwendet, um eine Semantik für Pläne, die auf dynamischer Aussagenlogik 1. Stufe basieren, zur Verfügung zu stellen. Die Semantik eines Programms wird in der dynamischen Aussagenlogik in Form von Aussagen über mögliche und notwendige Wahrheit von Prädikaten nach der Programmausführung dargestellt.

Ein Programm heißt hierbei total korrekt, wenn es (1) hält, und, (2) immer, wenn es hält, sind in der entstandenen Welt bestimmte Zielaussagen wahr. Programme, für die nur (2) zutrifft, heißen partiell korrekt. [3]

Rosenchein garantiert universelle Ausführbarkeit, indem er eine schleifenfreie Teilmenge der dynamischen Aussagenlogik wählt, die er weiter auf eine Menge sogenannter C-Programme beschränkt. Diese C-Programme schränken die Benutzung von Verzweigungen derart ein, daß sie das Halten des Planungsvorganges garantieren, egal welcher Zweig der Bedingung durchlaufen wird.

Temporale Projektion. Eine ähnliche Idee verfolgten Dean und Boddy [DB88] in ihrer Arbeit. Sie benutzen Aktionen, die variablenfreie Vorbedingungen und Effekte haben. Die Effekte der Schritte sind als Projektionsregeln gegeben und haben die Form $< e, \phi, \alpha, \delta >$. e ist hierbei ein Ereignis, das mit der Regel assoziert ist. ϕ ist eine Menge von vorausgehenden Vorbedingungen. Wenn diese Vorbedingungen wahr sind, werden die Bedingungen in α hinzugefügt; wenn sie falsch sind, werden die Bedingungen in δ gelöscht.

Dean und Boddy entwickelten algorithmische Näherungslösungen, die eine Untermenge der notwendigerweise wahren Aussagen berechnen, und eine Obermenge der Aussagen, die möglicherweises wahr sind. Die Komplexität dieser Algorithmen ist polynomial. Über die Qualität der Näherungslösungen ist jedoch nichts bekannt.

Desweiteren beschäftigten sich Dean und Boddy mit folgendem Entscheidungsproblem: Sei eine partiell geordnete Menge von Ereignissen $\mathcal{A}$ gegeben. Gehört eine Bedingung $\mathcal{C}$ zu $Possible(e)$? Letztere ist dabei eine Menge von Bedingungen, die in einer Vervollständigung von $\mathcal{A}$ direkt nach dem Ereignis e gelten.

In der Formulierung von Dean und Boddy ist $\mathcal{A}$ ausführbar, auch wenn die Vorbedingungen einer Regel nicht zutreffen. Die Regel hat dann keine Effekte (vgl. Abschnitt 4.3.2).

Daraus folgt, wie bereits mehrfach erläutert, daß mögliche Wahrheit äquivalent zu möglicher bedingter Wahrheit ist, und mögliche Wahrheit dual zu notwendiger Wahrheit ist. Dadurch sind Dean und Boddy in der Lage in ihrem Formalismus nachzuweisen, daß die Berechnung von möglicher Wahrheit NP-schwierig und die Berechnung von notwendiger Wahrheit co-NP-schwierig ist.

Bedingte Schritte. Chapman benutzt universell ausführbare Aktionen (er nennt sie bedingte Schritte), um sein *Intractability Theorem* für Aktionen mit bedingten Effekten zu beweisen. Er definiert in [Cha87, Seite 371]:

Ein *bedingter Schritt* ist jederzeit anwendbar, hat aber zwei Mengen von Nachbedingungen, zum eine if-true– und zum anderen if-false-Nachbedingungen. Die if-true-Nachbedingungen gelten in der Ausgabesituation, wenn alle Vorbedingungen in der Eingabesituation erfüllt sind; ansonsten gelten die if-false-Nachbedingungen.

Da bedingte Schritte jederzeit anwendbar sind, ist ein Plan, der nur aus solchen Schritten aufgebaut ist, immer ausführbar. Genau wie im Formalismus von Dean und Boddy ist die Bestimmung der notwendigen Wahrheit co-NP-schwierig, wie Chapman in seinem Beweis zum *Intractability Theorem* auch nachweist.

Da das *Intractability Theorem* auf Planungsoperatoren aufbaut, die bedingte Effekte haben, war es für Forscher im Planungsbereich nur natürlich, daß es an der Bedingtheit der Operatoren lag, warum notwendige Wahrheit so *widerspenstig* ist. Die Resultate dieses Theorems

[3] Kambhampati und Nau weisen an dieser Stelle auf die Ähnlichkeit von partieller Korrektheit und partieller Wahrheit hin.

hängen aber genauso von der universellen Ausführbarkeit von Chapmans bedingten Schritten ab, wie von deren Bedingtheit selbst. Dies soll nun erläutert werden.

Angenommen, $\mathcal{P}$ sei ein unvollständiger Plan, der aus unbedingten Schritten bestehe. Weiter sei a ein Schritt in $\mathcal{P}$, wobei $pre(a)$ und $post(a)$ eine freie Variable x enthalten. Dann sind die Effekte von a im Sinne der Planung und der temporalen Projektion in gewisser Weise bedingt. Genauer, a hat verschiedene Effekte in unterschiedlichen Vervollständigungen von $\mathcal{P}$, abhängig vom Wert, den x erhält.

Auf jeden Fall bleibt der Aufwand für die Berechnung notwendiger Wahrheit polynomial.

Chapmans Planungssprache enthält unendlich viele Konstantensymbole. Daraus folgt, daß immer eine Variablenbindung für x gefunden werden kann, die a nicht ausführbar macht. Als Konsequenz hat $\mathcal{P}$ immer mindestens eine Vervollständigung, die nicht ausführbar ist. Aus dieser Tatsache wiederum läßt sich schließen, daß die Berechnung notwendiger Wahrheit trivial ist: Nichts ist im Zielzustand von $\mathcal{P}$ notwendigerweise wahr.

Nun nehmen Nau und Kambhampati eine Einschränkung der Planungssprache $\mathcal{L}$ auf endlich viele Konstantensymbole an.[4] Dann gibt es einen Plan, in dem a ausführbar ist für jede Bindung von x. In diesem Fall wird die Prüfung auf notwendige Wahrheit co-NP-schwierig, auch ohne bedingte Schritte.

Satz 4: Wenn die Sprache $\mathcal{L}$ nur eine endliche Menge von Konstantensymbolen enthält, dann ist die Berechnung notwendiger Wahrheit co-NP-schwierig.

Dieses Resultat steht in direkter Beziehung zu Chapmans Beobachtung, daß die Einschränkung der Menge von Variablen auf eine endliche Menge das MTC untergräbt und Constraint-Berechnungen NP-vollständig macht.

Kohärente Pläne. Nebel und Bäckström [NB94] haben die Komplexität der Berechnung von Plangültigkeit und temporaler Projektion untersucht. Während die Untersuchungen von Kambhampati und Nau ursprünglich vom Fehlen modaler Dualität in Chapmans MTC motiviert war, haben sich Nebel und Bäckström für die Asymmetrie der temporalen Projektion und der Komplexität der Planvalidierung, wie sie Chapman bereits untersucht hat, interessiert.

Auch wenn Nebel und Bäckström ähnliche Resultate wie Kambhampati und Nau erhalten haben, weisen letztere jedoch auf signifikante Unterschiede hin. Nebel und Bäckström stützten sich nicht auf bedingte modale Wahrheit, sondern schränkten die Anwendbarkeit des MTC auf Pläne ein, deren Vervollständigungen alle ausführbar sind; sie nennen diese Eigenschaft der Ausführbarkeit *Kohärenz*. Weiterhin dürfen ihre Pläne keine Variablen enthalten, während die Pläne von Kambhampati und Nau keine solche Beschränkung verlangen. Sie sehen aber die Arbeit von Nebel und Bäckström als Komplement ihrer Forschungen.

[4] Dies impliziert eine endliche Menge von Grundtermen, da $\mathcal{L}$ funktionsfrei ist.

Kapitel 5

Das MTC im Planungssystem CAPlan

5.1 Das Planungssystem CAPlan

CAPlan[1] ist ein domänenunabhängiges, nichtlineares Planungssystem zur Unterstützung der Arbeitsplanerstellung bei der computerintegrierten Fertigung [Web94]. CAPlan verwendet, wie Kambhampati und Nau, eine STRIPS-artige Notation bei der Domänenmodellierung, arbeitet demnach mit einer zustandsorientierten Sicht der Welt. Diese Zustände entsprechen den *Situationen*, wie sie in Kapitel 2 vorgestellt wurden.

Planen wird im System CAPlan als Suche im Raum der Planzustände verstanden. Grundsätzlich liegt dem System der Planungsansatz der *systematischen nichtlinearen Planung (SNLP)* [MR91] zugrunde. Die Suche im Raum der möglichen Planzustände kann von beliebig definierbaren Kontrollkomponenten gesteuert werden, welche das Verhalten des Planers an Entscheidungspunkten festlegen. Die systematische Kontrollkomponente **SNLP+** basiert auf der Standardstrategie von SNLP, die als vollständig, korrekt und systematisch bekannt ist.

5.2 Planrepräsentation

Im System CAPlan bestehen Pläne aus einer Menge von Planschritten und einer auf diesen Schritten definierten (partiellen) Ordnung. Eine Menge von Variablenbindungs-Constraints (`Same/NotSame`) erlaubt Konsistenztests zur Prüfung der Übereinstimmungsrelation.

Zur Beschreibung von Situationen in der Planungswelt und zur Definition der STRIPS-Operatorschemata sind nur funktionsfreie Atomformeln zugelassen, d.h. Quantifizierungen sind nicht erlaubt. STRIPS-Operatoren sind definiert durch zwei Mengen: Zum einen durch eine Menge von *Vorbedingungen*, die erfüllt sein müssen, damit ein Operator anwendbar ist, und zum anderen durch eine Menge der durch den Operator bewirkten Effekte, die *Add-/Delete-Liste*. Die Operatoren werden in der Regel unter Verwendung von Variablen in Form der Operator-Schemata definiert (siehe [Web94]).

Da alle diese Bedingungen bereits gegeben waren, mußte nicht in das eigentliche Planungssystem eingegriffen werden, um das MTC zu realisieren. Die Implementation stellt somit eine

[1] <u>C</u>omputer <u>A</u>ided <u>Plan</u>ning

reine Erweiterung dar, eine zusätzliche Funktionalität bei Plan und Planschritt.

5.2.1 Planschritte

Allen Planschritten ist eindeutig ein Name und ein (STRIPS-)Operator zugeordnet. Durch diesen Operator sind jedem Planschritt eindeutig Vorbedingungen und Effekte, sowie Constraints über den eingeführten Variablen zugeordnet.

Es gibt zwei Arten von Planschritten, *normale* (Klasse `RealPlanStep`) und *künstliche* Schritte (Klasse `ArtificialPlanStep`). Von den künstlichen Planschritten existieren nur zwei Instanziierungen: `START` hat keine Vorbedingungen und entspricht Schritt s_0 aus Abschnitt 2.2; `FINISH` hat keine Effekte und beschreibt die aktuelle Problemstellung, deren Ziele Vorbedingungen von `FINISH` sind.

Die Unterscheidung in künstliche und normale Schritte hat keine Auswirkungen auf die Implementierung des MTC, da ihre Gemeinsamkeiten in einer Oberklasse zusammengefaßt sind (Klasse `PlanStep`).

5.2.2 Ordnungen

Betrachtet man eine Lösung (einen linearisierten Plan[2]) eines STRIPS-Planungsproblems, so muß für jeden korrekten Plan gelten, daß die Vorbedingungen aller vorkommenden Planschritte zur Ausführungszeit notwendigerweise erfüllt sind.

Eine Strategie, dies sicherzustellen, ist die Benutzung des MTC in der Goal-Achievement-Prozedur, wie sie Chapman im System TWEAK erfolgreich implementiert hat (vgl. Abschnitt 3.2).

Der SNLP-Ansatz geht Planungsprobleme auf eine andere Art an. Zu jeder Einführung eines Schrittes wird explizit der Zweck festgehalten, der zur Einführung des Schrittes geführt hat oder der den Schritt ausnutzt (phantomisiert). Realisiert werden diese Begründungen durch *Causal Links*, wodurch auch Interaktionen von Schritten untereinander erkennbar werden.

Die partielle Ordnung auf den Planschritten ist beim SNLP-Ansatz durch eine Menge von Ordnungsbeziehungen zwischen je zwei verschiedenen Schritten definiert. Es gibt zwei Arten von Ordnungen zwischen Schritten s und t, deren Hauptunterschied der Grund ihrer Einführung in den Plan ist. Beide ordnen s vor t an. Dies sind einerseits die bereits erwähnten Causal Links und zum anderen zusätzliche Ordnungen, *Protections*, die ggf. zur Auflösung von Interaktionen zwischen Schritten eingeführt werden.

Diese Ordnungen im Plan stellen die Grundlage für die Implementation des MTC in das Planungssystem CAPlan dar.

5.3 Protokolle

Ziel dieser Arbeit war es, für die Pläne in CAPlan ein MTC zu realisieren. Die im Rahmen dieser Implementierung entstandenen neuen Protokolle für den Plan bzw. seine Schritte sollen nun zusammengefaßt werden.

Das implementierte Wahrheitskriterium betrachtet den Ist-Zustand eines (unvollständigen) Plans und wird nicht als Planungsverfahren eingesetzt. Aus diesem Grund wird die

[2] In einem *linearisierten Plan* sind alle Planschritte total geordnet.

Ausführbarkeit der Schritte nicht weiter beachtet. Es wurde somit ein modales *bedingtes* Wahrheitskriterium realisiert, welches die von Kambhampati und Nau geforderten Bedingungen der Erzeugung und Nicht-Zerstörung (vgl. Abschnitt 4.1.1) erfüllt. Alle im folgenden gemachten Aussagen über Wahrheit von Prädikaten meint also die bedingte Wahrheit.

5.3.1 Erweiterung der Klasse `Plan`

Die Klasse `Plan` der Implementierung von CAPlan repräsentiert partiell geordnete Pläne des Systems und wurde um zwei Methoden bzgl. des MTC erweitert. Die beiden Methoden beschreiben die mögliche bzw. notwendige Situation an einem Schritt im Plan.

`necessarilyTrueAt: aPlanStep ofInputOnly: aBoolean` berechnet für den Planschritt `aPlanStep` die Menge der Prädikate, die vor oder nach der Ausführung dieses Schrittes *notwendigerweise wahr* sind. Ob `aPlanStep` bei der Berechnung berücksichtigt werden soll, wird durch `aBoolean` bestimmt.

`possiblyTrueAt: aPlanStep ofInputOnly: aBoolean` berechnet für den Planschritt `aPlanStep` die Menge der Prädikate, die vor oder nach der Ausführung dieses Schrittes *möglicherweise wahr* sind. Ob `aPlanStep` bei der Berechnung berücksichtigt werden soll, wird durch `aBoolean` bestimmt.

Zwei weitere Methoden liefern eine Liste bestehend aus einer Liste der notwendigerweise wahren und aus einer zweiten Liste der möglicherweise wahren Prädikate.

`inputSituationAt: aPlanStep` berechnet notwendigerweise und möglicherweise wahre Prädikate, die *vor* `aPlanStep` gültig sind.

`outputSituationAt: aPlanStep` berechnet notwendigerweise und möglicherweise wahre Prädikate, die *nach* `aPlanStep` gültig sind.

5.3.2 Erweiterung der Klasse `PlanStep`

Die oben beschriebene neue Funktionalität für den Plan baut auf den folgenden Methoden für MTC-Berechnungen der Klasse `Planschritt` auf.

`isNecessarilyTrue: aPredicate ofInputOnly: aBoolean` berechnet, ob `aPredicate` vor oder nach dem Schritt, der diese Botschaft gesandt wurde, *notwendigerweise wahr* ist. Auch hier bestimmt `aBoolean`, ob der angesprochene Schritt bzgl. seiner Effekte berücksichtigt werden soll, oder nicht.

`isPossiblyTrue: aPredicate ofInputOnly: aBoolean` berechnet, ob `aPredicate` vor oder nach dem Schritt, der diese Botschaft gesandt wurde, *möglicherweise wahr* ist. Auch hier bestimmt `aBoolean`, ob der angesprochene Schritt bzgl. seiner Effekte berücksichtigt werden soll, oder nicht.

`checkModalTruth: aPredicate ofInputOnly: aBoolean` wird von `isNecessarilyTrue` und `isPossiblyTrue` für die Berechnung aufgerufen. Diese Methode liefert genauere Rückgabewerte in Form eines Arrays:

- `#necessarilyTrue`, wenn das Prädikat notwendigerweise wahr ist, und dazu den erzeugenden Schritt,
- `#possiblyTrue`, wenn das Prädikat möglicherweise wahr ist, und dazu den eventuellen Clobberer,
- `#noEstablishment`, wenn das Prädikat nirgends zugesichert wurde,
- `#noDeclobbering`, wenn die Negation das Prädikats zugesichert wurde, und dazu den Clobberer.

Mit dem zuletzt aufgeführten Protokoll wäre es leicht möglich, Chapmans Goal-Achievement-Prozedur nachzubilden, da es alle dafür relevanten Informationen liefert.

5.3.3 Verwendung der Protokolle

Eine Verwendungsmöglichkeit der implementierten MTC-Methoden zeigt Abbildung 5.1. Sie zeigt einen unvollständigen Plan zum Sussman-Problem, bestehend aus den Schritten `START` und `FINISH`, sowie zwei zueinander parallel liegende Schritte `STEP-1` und `STEP-2`. Rechts daneben sind die Start- und die Zielsituation abgebildet.

Links unter dem Plan ist das Popup-Menü dargestellt, das sich in Verbindung mit dem gewählten Schritt (hier `STEP-1`) öffnet. Wird darin **`show input situation`** oder **`show output situation`** ausgewählt, erscheint eine Darstellung der mit Hilfe des MTC berechneten Ein- bzw. Ausgabesituation, wie neben der Menüdarstellung gezeigt. Die Menüpunkte

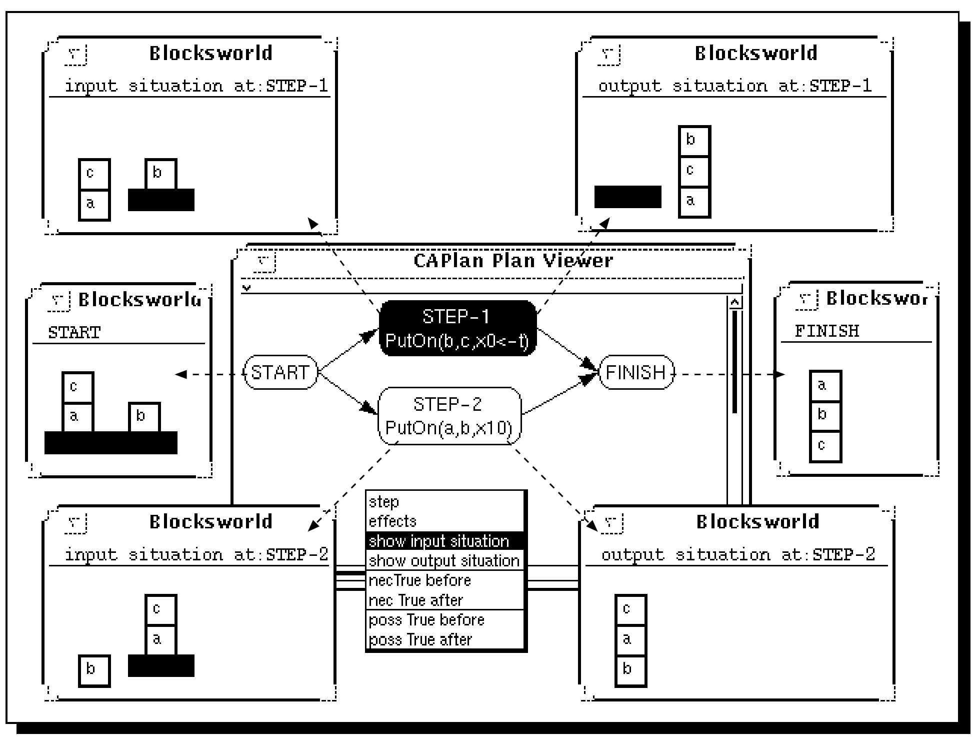


Abbildung 5.1: Ein- und Ausgabesituationen an Schritten.

`necTrueBefore` und `necTrueAfter` öffnen jeweils ein Fenster mit einer Liste aller notwendigerweise wahren Prädikate, die vor bzw. nach diesem Schritt gelten. Für die Punkte `possTrueBefore` und `possTrueAfter` gilt entsprechendes. Auch hier ist es wichtig zu beachten, daß die Ausführbarkeit der vor dem betrachteten Schritt liegenden Planschritte **nicht** unbedingt gewährleistet ist, somit nur notwendigerweise bzw. möglicherweise bedingt wahre Aussagen geliefert werden.

Literaturverzeichnis

[Cha85] D. Chapman. Nonlinear Planning: A Rigorous Reconstructions. Aus: *Proceedings of IJCAI-85*, Seite: 1022–1024, 1985.

[Cha87] D. Chapman. Planning for Conjunctive Goals. *Artificial Intelligence*, 32:333–337, 1987.

[DB88] Thomas Dean und Mark Boddy. Reasoning about partially oredered events. *Artificial Intelligence*, 36:375–399, 1988.

[ENS91] K. Erol, D.S. Nau und V.S. Subrahmanian. Complexity, Decidability and Undecidability Results for Domain-Independent Planning: A Detailed Analysis. Technischer Bericht CS-TR-2797, Computer Science Dept., University of Maryland, 1991.

[ENS92] K. Erol, D.S. Nau und V.S. Subrahmanian. When is Planning Decidable? Aus: *Proceedings of the 1st International Conference on AI Planning Systems (AIPS-92)*, Seite: 222–227, 1992.

[GR83] A. Goldberg und D. Robinson. *Smalltalk 80 - The Language and Its Implementation.* 2. Auflage. Addison-Wesley, Stuttgart, 1983.

[KN94] S. Kambhampati und D.S. Nau. On the Nature of Modal Truth Criteria in Planning. Aus: *Proceedings of AAAI-94*, Seite: 1055–1060, 1994.

[KN95] S. Kambhampati und D.S. Nau. On the Nature of Modal Truth Criteria in Planning. *Artificial Intelligence*, 1995. (to appear).

[KY95] C.A Knoblock und Q. Yang. Relating the Performance of Partial-Order Planning Algorithms to Domain Features. *SIGART Bulletin*, 6(1), 1995.

[McD91] D. McDermott. Regression Planning. *International Journal of Intelligent Systems*, 6:357–416, 1991.

[MR91] D. McAllester und D. Rosenblitt. Systematic Nonlinear Planning. Aus: *Proceedings of AAAI-91*, Seite: 634–639, 1991.

[NB94] B. Nebel und C. Bäckström. On the Computational Complexity of Temporal Projection, Planning and Plan Validation. *Artificial Intelligence*, 56(1), 1994.

[Pra] V. Pratt. Semantical Considerations on Floyd-Hoare Logic. Aus: *17th FOCS*, Seite: 109–121.

[Ric92] M.M. Richter. *Prinzipien der künstlichen Intelligenz.* 2. Auflage. B.G. Teubner, Stuttgart, 1992.

[Ros81] S. Rosenchein. Plan Synthesis: A Logical perspective. Aus: *IJCAI*, Seite: 331–337, 1981.

[Web94] F. Weberskirch. Realisierung eines nichtlinearen Planungssystems zur Unterstützung der Arbeitsplanerstellung bei der computerintegrierten Fertigung (CIM), 1994.

[Wil88] D.E. Wilkins. *Practical Planning - Extending the classical AI Planning Paradigm.* Morgan Kaufmann, 1988.

[YT90] Q. Yang und J.D. Tenenberg. Abtweak: Abstracting a Nonlinear, Least Commitment Planner. *AAAI-90*, Seite: 204–209, 1990.