

**Tableau-Kalkül für PDL
und Interpolation**

Manfred Borzechowski

Diplomarbeit,
angefertigt an der
FU Berlin, 1988

Inhalt

0	Einleitung	3
1	Der Tableau-Kalkül für PDL	
1.1	Die Syntax von PDL	4
1.2	Die Semantik von PDL	6
1.3	Entscheidungsverfahren für PDL .	10
1.4	Tableaus	14
1.5	Lokale Tableaus	18
1.6	PDL-Tableaus	24
1.7	Der PDL-Kalkül	29
1.8	Modellgraphen	31
1.9	Konsistente Knoten	34
1.10	Der Vollständigkeitssatz	36
1.11	Erweiterte PDL-Tableaus	38
2	Interpolation in PDL	
2.1	Der Interpolationssatz	39
2.2	Tableaus für Paare von Formelmengen	40
2.3	Konstruktion des Interpolanden .	42
2.5	Praktische Durchführung einer Konstruktion	55
2.6	Offene Fragen	59
	Anhang	
	Verzeichnis der Regeln	60
	Literaturverzeichnis	61

O Einleitung

Im Verlauf der letzten zwanzig Jahre wurden viele Logiken und Kalküle entwickelt und untersucht, die es erlauben, Eigenschaften von *Programmen* zu formalisieren und formal zu beweisen. Diesen Logiken gilt nicht nur theoretisches Interesse; vielmehr liefern sie die Voraussetzungen zur Lösung von Problemen der Programmverifikation und der automatischen Erzeugung von Programmen.

Zu diesen Logiken gehört die 1976 (PRATT [7]) definierte *dynamische Logik* oder *DL*; sie ist eine Verallgemeinerung der Modallogik 1.Stufe. Ihr aussagenlogisches Fragment, die *dynamische Aussagenlogik* oder *PDL* (für *propositional dynamic logic*), ist 1977 (FISCHER/LADNER [4]) eingeführt und als entscheidbar nachgewiesen worden. Seitdem beschäftigt sich nicht nur die Theoretical Computer Science, sondern auch die Mathematik mit DL, PDL und ihren Erweiterungen. Es ist zum Beispiel von großem Interesse, welche Eigenschaften der Modallogik auch diese Logiken aufweisen und in wie weit solche Eigenschaften mit für die Modallogik typischen Methoden bewiesen werden können. Eine umfassende Übersicht über die Arbeiten zu diesem Thema findet man im Literaturverzeichnis von HAREL [5] sowie in MÖLLER, RAUTENBERG [6].

Im ersten Teil der vorliegenden Arbeit wird ein Tableau-Kalkül für PDL entwickelt, der eine Weiterentwicklung des von PRATT [8] vorgeschlagenen Entscheidungsverfahrens darstellt. Im zweiten Teil wird mit Hilfe dieses Kalküls dann erstmals nachgewiesen, daß PDL die Interpolationseigenschaft hat.

1 Der Tableau-Kalkül für PDL

1.1 Die Syntax von PDL

In der Sprache der dynamischen Aussagenlogik gibt es zwei Arten von Ausdrücken: *Formeln* und *Programme*. Seien zunächst \mathcal{F}_0 und \mathcal{P}_0 zwei abzählbare Mengen, deren Elemente wir *Aussagenvariable* beziehungsweise *atomare Programme* nennen. Mit p, q, r, \dots bezeichnen wir die Aussagenvariablen und mit A, B, C, \dots die atomaren Programme. Aus diesen Mengen und aus den Zeichen $\emptyset, \neg, \wedge, \vee, \cup, *, ?, [,], ($ und $)$ sind die Formeln und Programme induktiv aufgebaut.

Definition. Jede Aussagenvariable sowie \emptyset ist eine Formel, und jedes atomare Programm ist ein Programm. Sind ferner P und Q beliebige Formeln und sind a und b beliebige Programme, so sind die Zeichenreihen

- $\neg P$, $(P \wedge Q)$ und $[a]P$ ebenfalls Formeln,
- $(a; b)$, $(a \cup b)$, a^* und $P?$ ebenfalls Programme.

Eine Zeichenreihe ist nur dann Formel oder Programm, wenn sie es schon auf Grund dieser Bildungsgesetze ist.

Damit ist zum Beispiel $(([A; B^*])p \wedge q)$ eine Formel. Im allgemeinen lassen wir äußere Klammerpaare von Formeln und Programmen weg; diese Formel würden wir also kürzer $[A; B^*]p \wedge q$ schreiben. Mit P, Q, R, \dots bezeichnen wir beliebige Formeln und mit a, b, c, \dots beliebige Programme. Weiter schreiben wir $P_1 \wedge \dots \wedge P_n$ für $(P_1 \wedge \dots (P_{n-1} \wedge P_n) \dots)$ und analog $a_1; \dots; a_n$ für $(a_1; \dots (a_{n-1}; a_n) \dots)$. Sind sogar alle a_i gleich einem gewissen a , so schreiben wir dafür a^n . Weiter sei $a^1 := a$ und $a^0 := \neg \emptyset?$. Wir benutzen auch die üblichen Abkürzungen \perp , $P \vee Q$, $P \rightarrow Q$ und $P \leftrightarrow Q$ für $\neg \emptyset$, $\neg(\neg P \wedge \neg Q)$, $\neg(P \wedge \neg Q)$ und $(P \rightarrow Q) \wedge (Q \rightarrow P)$.

Es bezeichnet \mathcal{F} die Menge aller Formeln und \mathcal{P} die Menge aller Programme. Mit X, Y, Z, \dots bezeichnen wir stets endliche Formelmengen. Dabei schreiben wir $X;Y$ für $X \cup Y$ und $X;P$ für $X \cup \{P\}$.

Eine Zeichenreihe $\{a\}$ nennen wir einen *Programmoperator*. Jedem Programmoperator $\{A\}$ mit $A \in \mathcal{P}_0$ fällt, wie wir noch sehen werden, in der Semantik von PDL eine Rolle zu, die der des Modaloperators in der modalen Logik K entspricht. Vielfach werden in der Literatur die Formeln von PDL auch mit Programmoperatoren der Form $\langle a \rangle$ aufgebaut. Diese sind die zu den Operatoren $\{a\}$ dualen Operatoren; man kann entweder $\{a\}P := \neg \langle a \rangle \neg P$ oder aber $\langle a \rangle P := \neg \{a\} \neg P$ für alle $a \in \mathcal{P}$ und $P \in \mathcal{F}$ definieren.

Programme der Gestalt $P?$ heißen *Tests*. Wegen dieser Tests können sowohl die Formeln als auch die Programme von PDL aus Formeln und Programmen aufgebaut sein. In der Formel $\{[B]P?; A^*I\}$ zum Beispiel taucht das Programm $\{B\}P?$ auf, das seinerseits aus der Formel $\{B\}P$ aufgebaut ist. Dieser verschachtelte Aufbau von Formeln und Programmen führt dazu, daß wir häufig folgende Beweismethode verwenden müssen:

Die simultane Induktion über den Formel- und Programmaufbau.

Es gelte die Eigenschaft E für alle $p \in \mathcal{F}_0$ sowie für 0 , und die Eigenschaft E' gelte für alle $A \in \mathcal{P}_0$. Weiter sei folgende Bedingung erfüllt:

Gilt E für P und Q , und gilt E' für a und b , so gilt E auch für $\neg P$, $(P \wedge Q)$, und $\{a\}P$, und E' gilt auch für $(a; b)$, $(a \cup b)$, a^* und $P?$.

Dann gilt E für alle $P \in \mathcal{F}$, und E' gilt für alle $a \in \mathcal{P}$.

1.2 Die Semantik von PDL

Wir stellen nun die Semantik von PDL vor. Sie ist eine Verallgemeinerung der relativistischen Semantik der Modallogik. Wir weisen darauf hin, daß für PDL auch eine algebraische Semantik existiert. Der an *dynamischen Algebren* interessierte Leser sei jedoch auf die in [6] zitierten Arbeiten von PRATT, KOZEN und REITERMAN/TRNKOVA verwiesen.

Definition. Ein *Kripke-Modell* ist eine Struktur der Gestalt (g, π, ν) . Dabei ist g eine nichtleere Menge, deren Elemente wir *Zustände* nennen und mit S, T, U, \dots bezeichnen. Weiter ist π eine Funktion, die jedem atomaren Programm A eine *Erreichbarkeitsrelation* $\pi(A) \subseteq g \times g$ zuordnet. Schließlich ist ν eine Funktion, die jeder Aussagenvariablen p eine Menge $\nu(p) \subseteq g$, ihren *Gültigkeitsbereich* zuordnet.

Sei nun $\mu = (g, \pi, \nu)$ ein Kripke-Modell fest gewählt. Wir zeigen, wie die Formeln und Programme von PDL über μ interpretiert werden.

Defintion. Wir definieren parallel die *Akzeptanzrelation* $\Vdash \subseteq g \times (\mathcal{F} \setminus \{0\})$ und die induktive Fortsetzung von π auf \mathcal{P} . Dabei schreiben wir $S \Vdash P$ für $(S, P) \in \Vdash$.

$S \Vdash p$	$\iff S \in \nu(p)$
$S \Vdash P \wedge Q$	$\iff S \Vdash P$ und $S \Vdash Q$
$S \Vdash \neg P$	\iff nicht $S \Vdash P$
$S \Vdash [a]P$	$\iff T \Vdash P$ für alle T mit $(S, T) \in \pi(a)$
$\pi(a \cup b)$	$= \pi(a) \cup \pi(b)$
$\pi(a ; b)$	$= \pi(a) \circ \pi(b)$
$\pi(a^*)$	$= \bigcup \{ \pi(a^i) \mid i \in \omega \}$
$\pi(P?)$	$= \{ (S, S) \mid S \Vdash P \}$

Wir bemerken, daß damit stets $S \Vdash 0$ und $(S, S) \in \pi(a^*)$ für alle $S \in g$ und $a \in \mathcal{P}$ gilt. Für $(S, T) \in \pi(a)$ schreiben wir häufig auch $S \rightarrow a T$.

Die Erreichbarkeitsrelation nichtatomarer Programme genügt also folgenden Beziehungen:

$$\begin{aligned} S-a;b \rightarrow T &\Leftrightarrow S-a \rightarrow U \text{ und } U-b \rightarrow T \text{ für ein } U \in \mathcal{G}, \\ S-a;u;b \rightarrow T &\Leftrightarrow S-a \rightarrow T \text{ oder } S-b \rightarrow T, \\ S-a^* \rightarrow T &\Leftrightarrow S-a^n \rightarrow T \text{ für ein } n \in \omega, \\ S-P? \rightarrow T &\Leftrightarrow S=T \text{ und } S \Vdash P. \end{aligned}$$

Ein Kripke-Modell kann als Modell eines Computers verstanden werden. Man stelle sich dazu \mathcal{G} als die Menge aller Zustände vor, die der Computer annehmen kann, definiert zum Beispiel durch die Menge aller möglichen Inhalte aller Speicher. Jedes $p \in \mathcal{G}$ repräsentiert dann eine gewisse Eigenschaft, die genau die Zustände in $v(p)$ aufweisen. Zu jedem atomaren Programm $A \in \mathcal{P}$, schließlich gehört ein gewisser Befehl, den das Rechenwerk ausführen kann. Es sind dann genau diejenigen $(S, T) \in \pi(A)$, bei denen T ein möglicher Endzustand der Ausführung von A , beginnend beim Zustand S ist. Es gilt genau dann $S \Vdash [A]p$, wenn $T \Vdash p$ gilt für jeden möglichen Endzustand dieser Ausführung. Durch diese Art der Interpretation eines Kripke-Modells können gewisse Programme durch typische Kontrollstrukturen imperativer Programmiersprachen wie PASCAL oder ALGOL interpretiert werden. Das ist zum Beispiel der Fall für die Programme

$$\begin{aligned} (P?;a) \cup (\neg P?;b) &\equiv \text{IF } P \text{ THEN } a \text{ ELSE } b, \\ a;(\neg P?;a)^*;P? &\equiv \text{REPEAT } a \text{ UNTIL } P, \\ (P?;a)^*; \neg P? &\equiv \text{WHILE } P \text{ DO } a. \end{aligned}$$

Wir wollen uns davon in einem Fall überzeugen. Es gilt $S-(\neg P?;a)^*;P? \rightarrow T \Leftrightarrow S-(\neg P?;a)^n \rightarrow T$ für ein n und $T \Vdash P$
 \Leftrightarrow es gibt S_i für $i < n$ mit
 $S = S_0 \rightarrow a \rightarrow \dots \rightarrow a \rightarrow S_n = T$ und $S_i \Vdash \neg P \Leftrightarrow i < n$,

und das ist genau dann der Fall, wenn T möglicher Endzustand der Ausführung von "REPEAT a UNTIL P " mit dem Anfangszustand S ist.

Definition. Gibt es für eine Formel P ein Kripke-Modell (g, v, π) und ein $S \in g$ mit $S \Vdash P$ (wir lesen "S akzeptiert P" oder "P gilt in S"), so heißt P *erfüllbar*. Weiter heißt P *gültig*, wenn $\neg P$ nicht erfüllbar ist.

Beispiel. Folgende Formeln sind gültig für alle P, Q, a, b:

$$\begin{array}{ll} [a](P \rightarrow Q) \rightarrow ([a]P \rightarrow [a]Q) & [a \cup b]P \leftrightarrow [a]P \wedge [b]P \\ [a^*]P \leftrightarrow (P \wedge [a][a^*]P) & P \wedge [a^*](P \rightarrow [a]P) \rightarrow [a^*]P \end{array}$$

Bemerkung. Wir wollen kurz einige Beziehungen von PDL zur Modallogik herstellen. Dazu bezeichnen wir mit $\mathcal{F}(a)$ die Menge aller Formeln von \mathcal{F} , in denen nur der Programmoperator $[a]$ auftaucht, und mit $\mathcal{F}'(a)$ bezeichnen wir die Menge aller gültigen Formeln von $\mathcal{F}(a)$. Jede Menge $\mathcal{F}'(a)$ ist im wesentlichen eine normale Modallogik (RAUTENBERG [11]); man muß lediglich den Modaloperator mit dem Programmoperator $[a]$ identifizieren. Einige bekannte Modallogiken erhält man bei geeigneter Wahl von a, so ist zum Beispiel:

$$K = \mathcal{F}'(A) \quad S4 = \mathcal{F}'(A^*) \quad M = \mathcal{F}'(A \cup 1?)$$

Die möglichen Erreichbarkeitsrelationen $\pi(A \cup 1?)$ und $\pi(A^*)$ aller Kripke-Modelle über eine Menge g sind nämlich gerade die reflexiven beziehungsweise reflexiven und transitiven Relationen auf g.

Die Logik CPDL ist eine Erweiterung von PDL, die durch Hinzunahme des Funktors $\bar{}$ entsteht (HAREL [5]), der durch $S \bar{a} \rightarrow T \Leftrightarrow T \rightarrow S$ interpretiert wird. Mit ihm ist:

$$B = \mathcal{F}'(A \cup 1? \cup A^-) \quad S5 = \mathcal{F}'((A \cup A^-)^*)$$

Die möglichen Erreichbarkeitsrelationen $\pi(A \cup 1? \cup A^-)$ und $\pi((A \cup A^-)^*)$ aller Kripke-Modelle über eine Menge g sind gerade die symmetrischen und reflexiven Relationen beziehungsweise die Äquivalenzrelationen auf g.

Definition. Sei X eine Formelmeng e und P eine Formel. Wenn für jedes Kripke-Modell (g, v, π) und jeden Zustand $S \in g$ aus $S \Vdash X$ bereits $S \Vdash P$ folgt, so schreiben wir $X \models P$ (wir lesen "aus X folgt P "). Ist $X = \{\emptyset\}$, schreiben wir auch $\emptyset \models P$. Gilt bereits $\emptyset \models P$, was genau dann der Fall ist, wenn P gültig ist, so schreiben wir einfach $\models P$.

Wir bemerken, daß in PDL kein Endlichkeitssatz für die Folgerungsrelation gilt. Sei zum Beispiel $X = \{[A^i]p \mid i \in \omega\}$ und $P = [A^*]p$. Dann gilt $X \models P$. Für jede echte Teilmenge $Y \subset X$ jedoch gilt $Y \not\models P$. Denn sei $[A^i]p \in Y$. In jedem Kripke-Modell $\mu = (g, v, \pi)$ mit $g = \omega$, $v(p) = \omega \setminus \{n\}$ und $\pi(A) = \{(i, i+1) \mid i \in \omega\}$ gilt $0 \Vdash Y$ und $0 \not\Vdash \neg [A^i]p$. Insbesondere gibt es also keine endliche Menge $X_0 \subset X$ mit $X_0 \models P$.

1.3 Entscheidungsverfahren für PDL

Eine der am frühesten erkannten Eigenschaften von PDL ist die in FISCHER, LADNER [4] bewiesene *small model property*.

Satz. Ist P erfüllbar, so ist P schon in einem Kripke-Modell (g, v, π) mit $|g| \leq 2^{l(P)}$ erfüllbar.

Dabei bedeutet $l(P)$ die Länge der Zeichenreihe P. Der Beweis des Satzes erfolgt mit der aus der Modallogik bekannten Technik der *Filtration*. Wir wollen die Beweisidee hier kurz skizzieren:

Beweis. Sei P erfüllbar und $l(P)=n$. Sei $c(P)$ die kleinste Menge, die P enthält, abgeschlossen gegenüber Subformelbildung ist und folgende Bedingungen erfüllt:

$$\begin{aligned} [a;b]REc(P) &\Rightarrow [a][b]REc(P) \\ [a \cup b]REc(P) &\Rightarrow [a]R, [b]REc(P) \\ [a^*]REc(P) &\Rightarrow [a][a^*]REc(P) \\ [Q?]REc(P) &\Rightarrow QEc(P). \end{aligned}$$

Man beweist durch Induktion, daß stets $|c(P)| \leq l(P)$ ist. Sei nun $\mu=(g, v, \pi)$ ein Kripke-Modell so, daß ein $S_p \in g$ existiert mit $S_p \Vdash P$. Für $S, T \in g$ sei

$$\begin{aligned} S \equiv_p T & :\Leftrightarrow (S \Vdash R \Leftrightarrow T \Vdash R \text{ für alle } REc(P)) \\ |S|_p & := \{T \mid T \equiv_p S\}. \end{aligned}$$

Wir definieren nun ein Kripke-Modell (g_p, v_p, π_p) mit $g_p := \{|S|_p \mid S \in g\}$. Wegen $|c(P)| \leq n$ ist $|g_p| \leq 2^n$. Mit den Definitionen

$$\begin{aligned} |S|_p \in v_p(p) & :\Leftrightarrow \text{es gibt } S' \in |S|_p \text{ mit } S' \in v(p) \\ (|S|_p, |T|_p) \in \pi_p(A) & :\Leftrightarrow \text{es gibt } S' \in |S|_p \text{ und } T' \in |T|_p \\ & \text{mit } (S', T') \in \pi(A) \end{aligned}$$

hat $\mu'=(g_p, v_p, \pi_p)$ folgende Eigenschaft:

$$\text{Für alle } REc(P) \text{ und } S \in g_p \text{ gilt } S \Vdash^* R \Leftrightarrow |S|_p \Vdash^* R$$

Damit gilt insbesondere $|S_p|_p \Vdash^* P$. \square

Aus diesem Satz läßt sich unmittelbar ein Entscheidungsverfahren für die Erfüllbarkeit einer Formel P

ableiten:

Untersuche alle Kripke-Modelle (g, v, π) mit (zum Beispiel) $g = \{1, \dots, K\}$, $K \leq 2^{1(P)}$ daraufhin, ob ein $S \in g$ mit $S \Vdash P$ existiert.

Dieses Verfahren ist jedoch ganz und gar unpraktikabel. Zwar ist der benötigte Aufwand für den Test, ob ein $S \in g$ mit $S \Vdash P$ existiert, nur ein Polynom in $|g| + 1(P)$ (FISCHER, LADNER [4]). Die Anzahl der zu untersuchenden Kripke-Modelle ist aber von der Größenordnung $2^{4^{1(P)}}$ (PRATT [8]), was eine praktische Durchführung des Verfahrens für andere als triviale Fälle ausschließt. Aus diesem Grund wurde stets nach besseren Entscheidungsverfahren gesucht, von denen wir eine Auswahl im folgenden kurz vorstellen wollen.

Bereits in PRATT [8] geschieht die Entscheidung der Erfüllbarkeit einer Formel P auf eine Art und Weise, wie sie auch bei uns erfolgt; und zwar durch einen *Tableau-Kalkül*. Die Erfüllbarkeit von P wird zunächst zurückgeführt auf das Vorhandensein gewisser Strukturen; bei PRATT werden dazu *filtered tableau closures* definiert, wir benutzen die direkte Verallgemeinerung der *model graphs* aus RAUTENBERG [12]. In beiden Fällen wird gezeigt, daß eine solche Struktur genau dann existiert, wenn es nicht möglich ist, gewisse Tableaus zu konstruieren. Die von uns definierten PDL-Tableaus sind ebenfalls eine direkte Verallgemeinerung der in RAUTENBERG [12] definierten Tableaus für die Modallogik. Wie dort ermöglichen sie uns hier außerdem den Beweis eines Interpolationsatzes.

Nach obigem Satz wird jede erfüllbare Formel P bereits von einem Zustand eines Kripke-Modells (g, v, π) akzeptiert, in dem jeder Zustand $S \in g$ eindeutig durch die Menge $\{R \in c(P) \mid S \Vdash R\}$ bestimmt ist. Mit den in PRATT [9] und HAREL [5] angegebenen Verfahren wird jeweils versucht, ein solches Modell direkt aus den Teilmengen von $c(P)$ zu konstruieren. Man bildet dazu (im 0. Schritt) die Menge g_0 derjenigen $Z \in c(P)$, die folgende Bedingungen erfüllen:

$$\begin{aligned}
\neg REc(P) &\Rightarrow (\neg REZ \Leftrightarrow REZ) \\
Q \wedge REc(P) &\Rightarrow (Q \wedge REZ \Leftrightarrow Q, REZ) \\
[a; b] REc(P) &\Rightarrow ([a; b] REZ \Leftrightarrow [a][b] REZ) \\
[aub] REc(P) &\Rightarrow ([aub] REZ \Leftrightarrow [a]R, [b] REZ) \\
[a^*] REc(P) &\Rightarrow ([a^*] REZ \Leftrightarrow [a][a^*]R, REZ) \\
[Q?] REc(P) &\Rightarrow ([Q?] REZ \Leftrightarrow (QEZ \Rightarrow REZ)).
\end{aligned}$$

Zu jedem in P auftauchenden atomaren Programm A wird dann $\pi_0(A)$ definiert durch

$$(Z, Z') \in \pi_0(A) \Leftrightarrow ([A] REZ \Rightarrow REZ')$$

Existiert nach dem i-ten Schritt ein $Z \in g_i$ und ein $[a] REc(P)$ so, daß $[a] REZ$ ist, jedoch gilt:

$$\text{für alle } Z' \text{ mit } (Z, Z') \in \pi_i(a) \text{ ist } REZ',$$

so bildet man $g_{i+1} := g_i \setminus \{Z\}$ und $\pi_{i+1} := \pi_i \cap (g_{i+1} \times g_{i+1})$. Nach höchstens $|g_0|$ Schritten liegt ein g_k vor, für das kein solches Z existiert. Ist g_k nichtleer, so gilt mit $Z \in v_k(p) \Leftrightarrow p \in Z$ für das Kripke-Modell $\mu = (g_k, v_k, \pi_k)$:

$$REc(P) \Rightarrow (Z \Vdash R \Leftrightarrow REZ).$$

Enthält also ein $Z \in g_k$ die Formel P, so ist P erfüllbar. Man kann insbesondere zeigen, daß P genau dann erfüllbar ist, wenn ein $Z \in g_k$ mit $p \in Z$ existiert. Somit bildet die oben angegebene Konstruktion von g_k ein Entscheidungsverfahren.

Weitere Arbeiten ([1] und [13]) geben Entscheidungsverfahren für die *deterministic propositional dynamic logic* DPDL an. In ihr werden die Formeln und Programme nur über *deterministische* Kripke-Modelle interpretiert; das sind Kripke-Modelle (g, v, π) , in denen jedes $\pi(A)$ eine partielle Funktion ist. Zu jedem $S \in g$ und $A \in P$ gibt es also höchstens ein $T \in g$ mit $S \xrightarrow{A} T$. Nicht jede erfüllbare Formel von PDL ist auch in einem solchen Modell erfüllbar; die Formel $\neg[A]p \wedge \neg[A]\neg p$ zum Beispiel ist nur in einem Zustand S erfüllt, zu dem zwei Zustände $S \xrightarrow{A} T, T'$ existieren mit $T \Vdash p$ und $T' \Vdash \neg p$.

Wie in VARDI, WOLPER [13] dargelegt wird, hat jede in einem deterministischen Kripke-Modell erfüllbare Formel P ein sogenanntes Baum-Modell. Das ist ein

Kripke-Modell (g, v, π) , bei dem die Struktur $(g, U, \pi(A))$ ein Baum ist (siehe 1.4). Weiter wird gezeigt, daß zu jeder Formel P ein Automat A_P der Größe $c^{l(P)^2}$ für gewisses c konstruiert werden kann, der genau die Baum-Modelle von P akzeptiert. Die Formel P ist also genau dann erfüllbar, wenn A_P irgend einen Baum akzeptiert. Verfahren, die dieses Problem entscheiden, sind bereits bekannt, der interessierte Leser sei jedoch auf die Literaturhinweise in [13] verwiesen.

1.4 Tableaus

In diesem Abschnitt definieren wir Tableaus und stellen die typischen Bestandteile von Tableau-Kalkülen vor.

Definition. Das Paar (g, \triangleleft) einer Menge g und einer Relation \triangleleft auf g heißt eine *Struktur*. Die Elemente s, t, \dots von g heißen *Knoten*; eine Menge s_0, \dots, s_n von Knoten mit $s_0 \triangleleft \dots \triangleleft s_n$ heißt *Pfad* (von s_0 zu s_n). Existiert ein Pfad von s zu t , schreiben wir $s \triangleleft t$; die Relation \triangleleft ist also der reflexive transitive Abschluß der Relation \triangleleft . Für $s \triangleleft t$ und $s \neq t$ schreiben wir $s < t$. Ist $s < t$, so heißt t *Nachfolger* von s , und s heißt *Vorgänger* von t . Existiert zudem kein $u \in g$ mit $s < u < t$, so sprechen wir vom *direkten* Nachfolger bzw. Vorgänger. Ein Knoten ohne Nachfolger heißt *Endknoten*.

Definition. Ein *Baum* ist eine Struktur (T, \triangleleft) , die folgende Eigenschaften hat:

- i) Es gibt genau einen Knoten $t_0 \in T$, der keinen Vorgänger hat. Dieser Knoten heißt die *Wurzel* von (T, \triangleleft) .
- ii) Für jeden Knoten $s \in T$ gibt es genau einen Pfad von t_0 zu s .

Wir nennen einen Baum (T', \triangleleft') *Teilbaum* von (T, \triangleleft) , wenn $T' \subseteq T$ und $\triangleleft' = \triangleleft|_{T'}$ ist, und wenn für $s, t \in T'$ mit $s < t$ jeder Knoten u mit $s < u < t$ ebenfalls in T' ist. Die *Länge* eines Baums (T, \triangleleft) ist die Anzahl der Knoten des längsten Pfades in (T, \triangleleft) .

Definition. Ein *Tableau* \mathcal{T} (über eine Formelmenge \mathcal{F}_T) ist eine Struktur der Gestalt $(T, \triangleleft, x, \mathcal{F}_T)$. Dabei ist (T, \triangleleft) ein endlicher Baum, und x ist eine Funktion, die jedem Knoten t eine endliche Formelmenge $x(t) \subseteq \mathcal{F}_T$ so zuordnet, daß stets folgende Bedingung gilt:

Ist $x(s)$ erfüllbar und s nicht Endknoten, dann ist $x(t)$ erfüllbar für mindestens einen direkten Nachfolger t von s .

Wir werden nicht immer streng zwischen der Struktur \mathcal{J} und der Knotenmenge T unterscheiden und von den Knoten $s, t, \dots \in \mathcal{J}$ reden. Außerdem werden wir, wo es nicht zu Mißverständnissen führen kann, nicht zwischen den Knoten und den ihnen zugeordneten Formelmengen unterscheiden. Wir werden also z.B. von einem Tableau "mit der Wurzel X " oder von "einem Tableau für X " reden, wenn seiner Wurzel die Formelmenge X zugeordnet ist.

Ein *Tableau-Kalkül* besteht aus zwei Komponenten:

- einer Menge von *Regeln*, die angeben, wie ein Tableau für X ausgehend von der Wurzel X durch schrittweise Bildung neuer Knoten konstruiert werden kann,
- einer einfach zu überprüfenden Eigenschaft von Tableaus, die die Nichterfüllbarkeit der Wurzel X impliziert. Tableaus mit dieser Eigenschaft nennen wir *geschlossen*.

Ein Tableau-Kalkül heißt *vollständig*, wenn man für jede nicht erfüllbare Formelmenge X mit den Regeln des Kalküls ein geschlossenes Tableau konstruieren kann. Ein vollständiger Kalkül eignet sich als Entscheidungsverfahren, wenn für jedes nicht erfüllbare X ein geschlossenes Tableau existiert, dessen Größe einen nur von $l(X) := \sum_{P \in X} l(P)$ abhängenden, berechenbaren Wert nicht überschreitet.

Bevor wir den Tableau-Kalkül für PDL vorstellen, wollen wir einen vollständigen Kalkül für das Problem angeben, ob eine Formel aus $\mathcal{F}(A)$ erfüllbar ist. Es handelt sich dabei im Prinzip um den Kalkül für die Modallogik K aus RAUTENBERG [12].

Definition. Eine Formelmenge X heißt *geschlossen*, wenn OEX ist oder wenn sie eine Formel zusammen mit ihrer Negation enthält. Sie heißt *einfach*, wenn jede Formel $P \in X$ eine (eventuell negierte) Aussagenvariable oder von der Gestalt $[A]R$ oder $\neg[A]R$ ist. Ferner sei $X_\Delta := \{R \mid [A]R \in X\}$.

Wir nennen dabei eine Formel P von der Gestalt $[A]R$, wenn es $A \in \mathcal{P}_0$ und $R \in \mathcal{F}$ gibt mit $P = [A]R$.

Lemma 1. Eine einfache Formelmengens X ist genau dann erfüllbar, wenn sie nicht geschlossen ist und für alle $\neg[A]R \in X$ auch $X_A; \neg R$ erfüllbar ist.

BEWEIS. Sei X erfüllbar. Dann gilt $S \Vdash X$ in einem Kripke-Modell $\mu = (g, \nu, \pi)$, und es gibt für alle $\neg[A]R \in X$ ein $S_{A,R} \in g$ mit $S \rightarrow A \rightarrow S_{A,R}$ und $S \Vdash X_A; \neg R$.

Sei andererseits X nicht geschlossen, und es gebe für jedes $\neg[A]R \in X$ ein Kripke-Modell $\mu_{A,R}$, in dem ein Zustand $S_{A,R}$ mit $S_{A,R} \Vdash X_A; \neg R$ existiert. Man fügt nun alle $\mu_{A,R}$ zusammen mit einem neuen Zustand S_0 zu einem weiteren Kripke-Modell zusammen. Seien o.B.d.A. die $g_{A,R}$ paarweise disjunkt, und sei S_0 in keinem $g_{A,R}$ enthalten. Sei $g := \{S_0\} \cup \bigcup (g_{A,R} \mid \neg[A]R \in X)$. Sei $\nu'(p) := \{S_0\}$, falls $p \in X$, sonst sei $\nu'(p) := \emptyset$ (hier geht ein, daß X nicht geschlossen ist). Sei $\nu(p) := \nu'(p) \cup \bigcup \{g_{A,R}(p)\}$. Sei weiter $\pi'(A) := \{(S_0, S_{A,R}) \mid \neg[A]R \in X\}$ und $\pi(A) := \pi'(A) \cup \bigcup \{\pi_{A,R} \mid \neg[A]R \in X\}$. In $\mu = (g, \nu, \pi)$ gilt dann $S_0 \rightarrow A \rightarrow S_{A,R}$ für alle $\neg[A]R \in X$. Man überzeugt sich nun schnell davon, daß $S_0 \Vdash X$. \square

Die Regeln des Kalküls lauten wie folgt:

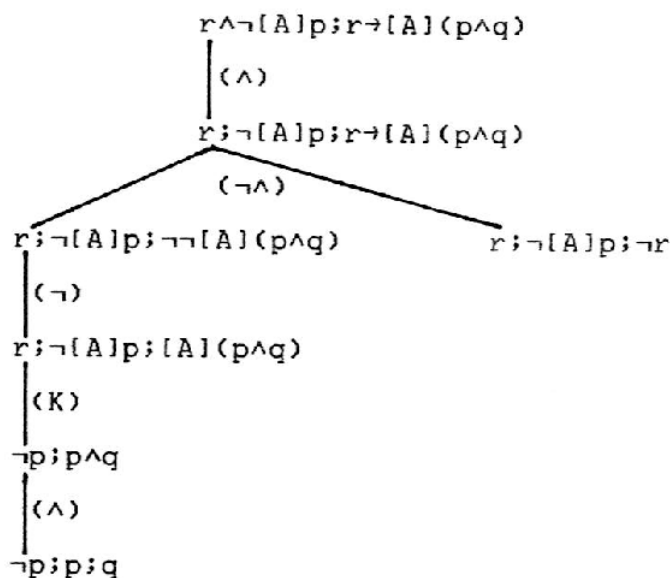
$$\begin{array}{lll}
 (\neg) \frac{X; \neg \neg P}{X; P} & (\wedge) \frac{X; P \wedge Q}{X; P; Q} & (\neg \wedge) \frac{X; \neg(P \wedge Q)}{X; \neg P \mid X; \neg Q} \\
 & (K) \frac{X; \neg[A]P}{X_A; \neg P} &
 \end{array}$$

Die Schreibweise der Regeln entspricht der in [12] und ist so zu verstehen: die Nachfolger eines Knotens Z können dann mit der Regel $(\neg \wedge)$ konstruiert werden, wenn er von der Gestalt $X; \neg(P \wedge Q)$ ist, d.h. wenn es $X \in \mathcal{F}$ und $P, Q \in \mathcal{F}$ gibt mit $Z = X; \neg(P \wedge Q)$. Der Knoten Z erhält dadurch die zwei Nachfolger $X; \neg P$ und $X; \neg Q$. Diesen Vorgang nennen wir eine Anwendung der Regel $(\neg \wedge)$ auf Z (oder auf $\neg(P \wedge Q)$ in Z). Die Anwendung der restlichen Regeln ist analog zu interpretieren. Auf jeden Knoten darf dabei nur einmal eine Regel angewandt werden. In diesem Kalkül nennen wir ein Tableau \mathcal{T} geschlossen, wenn jeder Endknoten von \mathcal{T} geschlossen ist.

Ein geschlossenes Tableau \mathcal{J} für ein nicht erfüllbares Z konstruiert man schrittweise wie folgt: Sei X ein bereits konstruierter, nicht erfüllbarer Knoten des Tableaus, der nicht schon geschlossen ist. Ist X nicht einfach, kann auf ihn eine der Regeln (\wedge) , $(\neg\wedge)$ oder (\neg) angewandt werden. Dadurch erhält X einen oder zwei direkte Nachfolger, und diese Knoten sind ebenfalls nicht erfüllbar. Ist X dagegen einfach, so gibt es nach Lemma 1 ein $\neg[A]R\text{EX}$ so, daß $X_A; \neg R$ nicht erfüllbar ist. Durch Anwendung der Regel (K) kann man aber genau diesen Knoten konstruieren.

Jeder so konstruierte Nachfolger Y eines nicht erfüllbaren Knotens X ist ebenfalls nicht erfüllbar, zudem gilt $l(Y) < l(X)$. Man erhält also schließlich ein Tableau, das nicht länger als $l(Z)$ ist, dessen nicht erfüllbare Endknoten alle geschlossen sind.

Beispiel. Wir geben ein geschlossenes Tableau für die Menge $\{r \wedge \neg[A]p, r \rightarrow [A](p \wedge q)\}$ an. Nach jedem Knoten haben wir dabei die Bezeichnung der Regel notiert, die zur Konstruktion seiner Nachfolger benutzt wurde.



1.5 Lokale Tableaus

Wie wir im letzten Abschnitt an einem Beispiel gesehen haben, kann man für jede Formelmenge $X \subseteq \mathcal{F}(A)$ problemlos ein Tableau konstruieren, dessen Endknoten X_1, \dots, X_n einfach sind und

$$S \Vdash X \Leftrightarrow S \Vdash X_i \text{ für ein } i \leq n.$$

für jeden Zustand S jedes Kripke-Modells μ gilt. Wir wollen nun zeigen, wie man auch für jedes $X \subseteq \mathcal{F}$ ein Tableau mit einer vergleichbaren Eigenschaft konstruieren kann. Dazu müssen wir die Menge der "Formeln" \mathcal{F} , aus denen die Knoten eines solchen Tableaus bestehen können, über \mathcal{F} hinaus erweitern.

Definition. Eine *n-Formel* sei eine Zeichenreihe, die aus einer Formel durch Ersetzen eines oder mehrerer Programmoperatoren der Gestalt $[a^*]$ durch die Zeichenreihe $[a^{(n)}]$ hervorgeht. Die Funktion f ordne jeder *n-Formel* diejenige Formel $P \in \mathcal{F}$ zu, aus der sie so hervorgeht; und für alle $P \in \mathcal{F}$ sei $f(P) := P$. Sei \mathcal{F}^n die Menge aller *n-Formeln*. Wir bezeichnen auch *n-Formeln* mit den Buchstaben P, Q, \dots und nennen sie ebenfalls *Formeln*. Um zu betonen, daß eine Formel keine *n-Formel* ist, nennen wir sie auch *normal*. Wir definieren für alle $P \in \mathcal{F}^n$

$$S \Vdash P \Leftrightarrow S \Vdash f(P).$$

Damit ist auch klar, was unter der Erfüllbarkeit einer Menge $X \subseteq \mathcal{F} \cup \mathcal{F}^n$ zu verstehen ist. Unter einer Formelmenge verstehen wir ab jetzt stets eine (endliche) Menge $X \subseteq \mathcal{F} \cup \mathcal{F}^n$. Wir nennen einen Knoten *normal*, wenn er nur normale Formeln enthält, sonst nennen wir ihn auch einen *n-Knoten*. Einen Knoten, der eine Formel der Gestalt $\neg[a^{(n)}]P$ enthält, nennen wir einen $\neg[a^{(n)}]$ -Knoten.

Definition. Ein *lokales Tableau* ist ein Tableau, dessen Wurzel eine normale Formelmenge ist, und das mit den klassischen Regeln

$$(\neg) \frac{X; \neg\neg P}{X; P}$$

$$(\wedge) \frac{X; P \wedge Q}{X; P; Q}$$

$$(\neg\wedge) \frac{X; \neg(P \wedge Q)}{X; \neg P \mid X; \neg Q}$$

sowie mit den folgenden Regeln konstruiert wird:

$$(\neg u) \frac{X; \neg[aub]P}{X; \neg[a]P \mid X; \neg[b]P} \quad (\neg?) \frac{X; \neg[Q?]P}{X; Q; \neg P} \quad (\neg;) \frac{X; \neg[a;b]P}{X; \neg[a][b]P}$$

$$(u) \frac{X; [aub]P}{X; [a]P; [b]P} \quad (?) \frac{X; [Q?]P}{X; \neg Q \mid X; P} \quad (;) \frac{X; [a;b]P}{X; [a][b]P}$$

$$(\neg n) \frac{X; \neg[a^*]P}{X; \neg P \mid X; \neg[a][a^{(n)}]P} \quad (n) \frac{X; [a^*]P}{X; P; [a][a^{(n)}]P}$$

Bei der Konstruktion sind zusätzlich folgende Sonderregelungen und Einschränkungen zu beachten:

- 1) Anstelle eines Knotens $X; \neg[A]P$ oder $X; [A]P$ mit der n -Formel P entsteht stets der Knoten $X; \neg[A]f(P)$ bzw. $X; [A]f(P)$.
- 2) Anstelle eines Knotens $X; [a^{(n)}]P$ entsteht stets der Knoten X .
- 3) Wenn es möglich ist, muß eine Regel auf eine n -Formel angewandt werden.
- 4) Auf einen $\neg[a^{(n)}]$ -Knoten darf keine Regel angewandt werden.

Entstehen aus der Menge X durch Anwendung einer der Regeln die Mengen X_1, \dots, X_n ($n \leq 2$), so gilt auch hier stets $SIH^*X \Leftrightarrow SIH^*X_i$ für ein $i \leq n$. Außerdem gilt, daß jede Formel durch Anwendung einer Regel durch in gewissem Sinne einfachere Formeln ausgetauscht wird; das sind z.B. Formeln, die kürzer sind oder mit einem einfacheren Programmoperator beginnen. Eine normale Menge, auf die keine dieser Regeln angewandt werden kann, ist einfach.

Definition. Ein lokales Tableau heißt *maximal*, wenn auf keinen Endknoten eine Regel angewandt werden kann oder darf.

Dort, wo in einem lokalen Tableau ein n -Knoten auftritt, befindet er sich in einem Teiltabelleau, welches nur aus n -Knoten besteht, und dessen Endknoten entweder $\neg[a^{(n)}]$ -Knoten sind oder normale Nachfolger auf Grund von 1) oder 2) haben. Wir geben ein Beispiel.

Sei nun $X_1 \in \dots \in X_n$ ein Pfad eines lokalen Tableaus \mathcal{T} , das so konstruiert ist, daß (mit den durch 3) gegebenen Einschränkungen) in jedem Knoten X_i ($i < n$) eine Formel P_i mit maximalem $m(P_i)$ ausgetauscht wird. Wir zeigen:

Für $j > i$ ist $P_i \notin X_j$.

Sei P_i normal. Es gilt $m(P_j) \leq m(P_i)$ für alle $j \geq i$. Aus keinem P_j kann P_i entstehen, somit ist $P_i \notin X_j$ für $j > i$. Sei P_i n -Formel und X_i erster normaler Knoten nach X_1 . Für $i \leq j < l$ gilt ebenfalls $m(P_j) \leq m(P_i)$, also ist $P_i \notin X_j$ für $i < j \leq l$. Ist P_i nichtnegierte n -Formel, so ist P_i von der Gestalt $[a_1] \dots [a_n] [a^*] Q$ mit einem normalen Q , und es ist $P_k = [a^*] Q$ für ein $k < i$. Da $[a^*] Q \in X_j$ für $j > k$ gilt, und P_i nur aus $[a^*] Q$ entstehen kann, gilt $P_i \in X_j$ auch für alle $j > l$. Analoges gilt für negierte n -Formeln.

Entlang des Pfades wird also keine Formel mehr als einmal ausgetauscht, in ihm tauchen höchstens $m(X_1)$ Formeln auf, er ist also nicht länger als $l(X_1)$. \square

Lemma 3. Sind s, t Knoten eines lokalen Tableaus mit $s < t$, so ist $x(s) \neq x(t)$.

BEWEIS. Eine Formel P wird bei Anwendung einer Regel gegen (höchstens) zwei Formeln P_1, P_2 ausgetauscht.

Wegen $m(P) > m(P_i)$ ($i=1,2$) gilt mit dem Maß $m'(X) := \sum_{P \in X} 3^{m(P)}$ stets $m'(x(s)) > m'(x(t))$. \square

Lemma 4. Sei μ ein Kripke-Modell über g , und sei $S \in g$. Sei $X = X'$; $\neg(a)P$ Knoten eines lokalen Tableaus mit $S \Vdash X$. Sei X' normal, und sei P normal falls a atomar ist; sonst sei P n -Formel. Weiter sei $T \in g$ mit $S \xrightarrow{a} T$ und $T \Vdash \neg P$. Dann gibt es einen normalen Knoten $Y \geq X$ mit $S \Vdash Y$, der die Formel $\neg([a_1] \dots [a_n] f(P))$ für ein $n \in \omega$ enthält. Ist $n > 0$, so ist a_i ein atomares Programm und $S \xrightarrow{a_i} \dots \xrightarrow{a_n} T$ gilt. Ist $n=0$, so ist $S=T$.

BEWEIS. Durch Induktion über a .

$a=A$: Die Behauptung gilt mit $Y=X$.

$a=Q?$: Es gibt nur einen Zustand T mit $S \xrightarrow{Q} T$, nämlich

$T=S$. Der direkte Nachfolger von X ist $X'; Q; \neg f(P)$, er ist normal und gilt in S .

$a=buc$: Es ist $S-b \rightarrow T$ oder $S-c \rightarrow T$. Ist o.B.d.A. $S-b \rightarrow T$, so gilt der Nachfolger $X'; \neg[b]P$ von X in S . Anwenden der Induktionsvoraussetzung liefert die Behauptung.

$a=b;c$: Es gibt UEg mit $S-b \rightarrow U-c \rightarrow T$. In U gilt $\neg[c]P$, somit kann man auf den Nachfolger $X'; \neg[b][c]P$ die Induktionsvoraussetzung anwenden. Es existiert also $Z \succ X$ mit $S \Vdash Z$ und Z enthält die n -Formel $\neg[a_1]..[a_n][c]P$. Ist $n > 0$, sind wir fertig, sonst wenden wir auf Z noch einmal die Induktionsvoraussetzung an.

$a=b^*$: Ist $S=T$, so gilt der Nachfolger $X'; \neg P$ in S . Ist $S \neq T$, so gibt es UEg mit $U \neq S$ und $S-b \rightarrow U-b^* \rightarrow T$ und $U \Vdash \neg[b^*]P$. Somit kann man auf den Nachfolger $X'; \neg[b][b^{(n)}]P$ die Induktionsvoraussetzung anwenden. Wegen $S \neq U$ existiert ein Nachfolger dieses Knotens, der in S gilt und der eine n -Formel der Gestalt $\neg[a_1]..[a_n][b^{(n)}]P$ enthält, wobei a_i atomar ist. \square

Lemma 5. Sei \mathcal{J} ein lokales Tableau, in dem jeder Endknoten normal oder ein $\neg[a^{(n)}]$ -Knoten ist. Seien X_1, \dots, X_n die normalen Endknoten. Sei S Zustand eines Kripke-Modells μ über g . Es gilt

$$S \Vdash^* X \iff S \Vdash^* X_i \text{ f\u00fcr ein } i \leq n$$

BEWEIS. Da mit $Y \triangleleft Z$ und $S \Vdash^* Z$ auch $S \Vdash^* Y$ gilt, folgt aus $S \Vdash^* X_i$ sofort $S \Vdash^* X$. F\u00fcr die andere Richtung zeigen wir, da\u00df zu jedem normalen Knoten $Y \in \mathcal{J}$ mit $S \Vdash^* Y$ ein normaler Knoten $Z \succ Y$ mit $S \Vdash^* Z$ existiert. Wir brauchen dazu nur die zwei F\u00e4lle zu betrachten, in denen nicht alle direkten Nachfolger von Y normal sind:

- Auf jeden Knoten, der eine nichtnegierte n -Formel enth\u00e4lt, kann eine Regel angewandt werden. Entsteht der Nachfolger von Y durch Anwendung von (n) , findet man also einen Pfad in S erf\u00fcllbarer Knoten, der von Y zu einem normalen Knoten Z f\u00fchrt.

- Sei $Y=Y'; \neg[a^*]P$, und die Nachfolger von Y entstehen durch Austausch von $\neg[a^*]P$ gemäß ($\neg n$). Ist a atomar, so sind schon beide Nachfolger von Y normal. Sei a also nicht atomar. Es gibt einen Zustand $U \in \mathcal{G}$ mit $S \xrightarrow{a^*} U$ und $U \Vdash \neg P$:
 Ist $S=U$, so ist schon der Nachfolger $Y'; \neg P$ von Y in S erfüllbar und normal.
 Ist $S \neq U$, so gibt es T mit $T \neq S$ und $S \xrightarrow{a} T \xrightarrow{a^*} U$ und $T \Vdash \neg[a^*]P$. Anwendung von Lemma 4 auf $Y; \neg[a][a^*]P$ und T liefert einen in S erfüllbaren normalen Knoten $Z \succ Y$. \square

1.6 PDL-Tableaus

Die Formelmengemenge X enthalte eine Formel der Gestalt $\neg[a_1]..[a_n]P$. Notwendige Bedingung dafür, daß $S \Vdash X$ in einem Kripke-Modell über g gilt, ist die Existenz von Zuständen $T_1, \dots, T_n \in g$ mit $S \xrightarrow{a_1} T_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} T_n$ und $T_n \Vdash \neg P$. Wir geben in diesem Abschnitt die Regeln der *PDL-Tableaus* an, mit denen für X diese Bedingung überprüft werden kann. Dazu gehört eine Regel, durch deren Anwendung eine Formel der Gestalt $\neg[a_1]..[a_n]P$ mit der Formel P *markiert* wird. Wir schreiben dann $\neg[a_1]..[a_n]P^*$. Enthält eine Formelmengemenge eine markierte Formel, nennen wir sie *belastet*, sonst *frei*. Analog sprechen wir von belasteten und freien Knoten.

Die Regeln der PDL-Tableaus sind neben den Regeln der lokalen Tableaus:

- (M+) $\frac{X; \neg[a_0]..[a_n]P}{X; \neg[a_0]..[a_n]P^*}$ X frei die *Belastungsregel*,
- (M-) $\frac{X; \neg[a]P^*}{X; \neg[a]P}$ die *Befreiungsregel*,
- (At) $\frac{X; \neg[A]P^*}{X_A; \neg P^{R^*P}}$ die *kritische Regel*.

Hierbei bedeute

$$\neg P^{R^*P} := \begin{cases} \neg P & \text{falls } R=P \\ \neg P^* & \text{sonst} \end{cases}$$

Die Regeln $(\neg\cup)$, $(\neg\cap)$, $(\neg;)$ und $(\neg?)$ der lokalen Tableaus sollen auch auf markierte Formeln anwendbar sein. Die jeweils entstehenden Formeln müssen dann auf eine bestimmte, noch festzulegende Weise markiert sein. Wir geben diese Regeln also noch einmal für die Anwendung auf markierte Formeln an.

- ($\neg\cup$) $\frac{X; \neg[a\cup b]P^*}{X; \neg[a]P^* \mid X; \neg[b]P^*}$ ($\neg;$) $\frac{X; \neg[a;b]P^*}{X; \neg[a][b]P^*}$
- ($\neg\cap$) $\frac{X; \neg[a^*]P^*}{X; \neg P^{R^*P} \mid X; \neg[a][a^{(n)}]P^*}$ ($\neg?$) $\frac{X; \neg[Q?]P^*}{X; Q; \neg P^{R^*P}}$

Man bemerke, daß bei Anwendung der Regeln (At) , $(\neg\cap)$

und $(\neg?)$ auf eine markierte Formel die Markierung genau dann verschwindet, wenn die entstehende Formel die Negation der Markierung ist.

Definition. Sei $X_1 \triangleleft \dots \triangleleft X_n$ ein Pfad eines mit den bisher definierten Regeln konstruierten Tableaus. Entsteht kein X_i ($i > 1$) durch Anwendung von (At) , nennen wir den Pfad *unkritisch*, sonst *kritisch*. Ist jedes X_i belastet, nennen wir den Pfad ebenfalls belastet.

Definition. Ein *PDL-Tableau* ist ein mit den bisher definierten Regeln konstruiertes Tableau, bei dessen Konstruktion zusätzlich zu 1) bis 4) folgende Sonderregelungen und Einschränkungen beachtet sind:

- 5) Auf einen Knoten, der durch Anwendung von $(M+)$ entsteht, darf nicht $(M-)$ angewandt werden.
- 6) Ein normaler Knoten $X=x(t)$, zu dem ein $s < t$ mit $x(s)=x(t)$ existiert, ist Endknoten, wenn der Pfad von s zu t kritisch ist, und wenn der Pfad von s zu t zudem belastet ist, wenn X belastet ist.
- 7) Jeder belastete Knoten, der nicht auf Grund von Bedingung 6) Endknoten ist, hat einen Nachfolger.

Ein *lokales PDL-Tableau* ist ein PDL-Tableau, in dem kein Knoten durch (At) entsteht.

Zu jedem PDL-Tableau \mathcal{J} gehört auf natürliche Weise eine Partition in lokale PDL-Tableaus $\mathcal{J}_1, \dots, \mathcal{J}_n$, und zwar in die maximalen lokalen Teilttableaus. Aus jedem normalen Endknoten eines \mathcal{J}_i , der nicht schon Endknoten von \mathcal{J} ist, entsteht durch Anwendung von (At) die Wurzel eines $\mathcal{J}_j \neq \mathcal{J}_i$. Ein lokales PDL-Tableau unterscheidet sich von einem lokalen Tableau nur durch die eventuelle Anwendung der Regeln $(M+)$ und $(M-)$, sie haben also weitgehend gleiche Eigenschaften. Wir lassen deswegen, wo es nicht zu Mißverständnissen kommen kann, den Zusatz PDL weg. Wir reden auch einfach von *Tableaus*, wenn wir PDL-Tableaus meinen.

Lemma 6. Es gibt eine Höchstlänge für Tableaus für X.

BEWEIS. Zunächst beweist man ähnlich wie in Lemma 2, daß in einem Tableau \mathcal{J} für X nicht mehr als $n \cdot l(X)$ verschiedene nicht markierte Formeln auftreten können. Weiter folgt aus Lemma 3 für alle Knoten $s, t \in \mathcal{J}$ mit $x(s) = x(t)$, daß der Pfad von s zu t kritisch ist. Kein Pfad enthält deshalb mehr als 2^n freie normale Knoten. Weiter gibt es in \mathcal{J} nicht mehr als $n \cdot 2^n$ verschiedene belastete Mengen für jede mögliche Markierung. Ein belasteter Pfad enthält also höchstens $n \cdot 2^n$ normale Knoten. Befinden sich zwischen zwei solcher Knoten nur n-Knoten, so sind es weniger als 2^n n-Knoten, denn sie befinden sich dann in demselben unkritischen Pfad. Ein belasteter Pfad ist also nicht länger $n \cdot 2^{2^n}$. Zwischen zwei belasteten Pfaden befindet sich mindestens ein freier Knoten, somit können nur weniger als 2^n belastete Pfade aufeinander folgen. Das Tableau \mathcal{J} ist also nicht länger als $n \cdot 2^{4^n}$. \square

Definition. Ist \mathcal{J} ein Tableau, so sei für $s, t \in \mathcal{J}$

$s \triangleleft^* t \iff s \triangleleft t$, oder s ist Endknoten, es gibt ein $u \triangleleft s$ mit $u \triangleleft t$ und $x(u) = x(s)$, von dem ein belasteter und kritischer Pfad zu s führt.

Weiter sei \triangleleft^* die reflexive transitive Hülle von \triangleleft^* . Die Relation ist i.a. keine Halbordnung; für freie Knoten X, Y gilt jedoch $X \triangleleft Y \iff X \triangleleft^* Y$. Eine Folge $t_1 \triangleleft^* \dots \triangleleft^* t_n$ nennen wir einen \triangleleft^* -Pfad.

Lemma 7. Sei S Zustand eines Kripke-Modells über g, und sei $X = X'; \neg[a]P^R$ ein in S geltender normaler Knoten eines Tableaus \mathcal{J} , in dem kein Knoten durch Anwendung von (M-) entsteht. Sei weiter $T \in g$ mit $S \xrightarrow{a} T$ und $T \Vdash \neg P$. Dann gibt es einen \triangleleft^* -Pfad erfüllbarer Knoten von X zu einem normalen Knoten $Y = Y'; \neg P^R$, der in T gilt.

BEWEIS. Wir bemerken zuerst, daß in \mathcal{J} jeder belastete Knoten s Nachfolger bezüglich \triangleleft^* besitzt, die aus Anwendung einer Regel auf $x(s)$ entstehen. Dann gibt es einen \triangleleft^* -Pfad erfüllbarer Knoten von X zu einem in S geltenden Knoten $Z = Z'; \neg[a]P^R$, dessen Nachfolger durch Austausch der markierten Formel entsteht. Der weitere Beweis erfolgt nun durch Induktion über a.

Induktionsanfang:

a=A: Sei $T \in \mathcal{G}$ mit $S \rightarrow T$ und $T \Vdash \neg P$. Der \leftarrow^* -Nachfolger $Z'; \neg P^{R^*}$ von Z gilt in diesem T und ist von der verlangten Gestalt.

a=Q?: Es ist $T=S$. Der \leftarrow^* -Nachfolger $Z'; Q; \neg P^{R^*}$ von Z gilt also in T und ist von der verlangten Gestalt.

Sei nun die Behauptung bewiesen für die Programme b und c .

a=buc: Sei $S \rightarrow buc \rightarrow T$ und $T \Vdash \neg P$. Dann ist $S \rightarrow b \rightarrow T$ oder $S \rightarrow c \rightarrow T$, und einer der \leftarrow^* -Nachfolger $Z'; \neg [b]P^R$ und $Z'; \neg [c]P^R$ von Z gilt in S . Anwenden der Induktionsvoraussetzung auf diesen Knoten liefert die Behauptung.

a=b;c: Für $S \rightarrow b;c \rightarrow T$ existiert U mit $S \rightarrow b \rightarrow U \rightarrow c \rightarrow T$ und $U \Vdash \neg [c]P$. Anwenden der Induktionsvoraussetzung auf den \leftarrow^* -Nachfolger $Z'; \neg [b][c]P^R$ von Z und U liefert einen Knoten $W=W'; \neg [c]P^R$ mit $Z \leftarrow^* W$, der in U gilt. Anwenden der Induktionsvoraussetzung - auf W und T liefert die Behauptung.

a=b*: Ist $S \rightarrow a^* \rightarrow T$, so gibt es ein minimales n mit $S \rightarrow a^n \rightarrow T$. Wir beweisen diesen Fall durch Induktion über n .

Induktionsanfang $n=0$:

Ist $n=0$, so ist $S=T$, der \leftarrow^* -Nachfolger $Z'; \neg P^{R^*}$ von Z gilt in S , und er ist von der gewünschten Gestalt.

Sei die Behauptung bewiesen für $m=n-1$:

Ist $n>0$, so gibt es $U \in \mathcal{G}$ mit $U \neq S$ und $S \rightarrow b \rightarrow U \rightarrow b^* \rightarrow T$. Der \leftarrow^* -Nachfolger $Z'; \neg [b][b^{(n)}]P$ gilt in S . Nach Lemma 4 gibt es einen normalen Knoten $W \geq^* Z$ mit $W=W'; \neg [a_1] \dots [a_k][b^*]P^R$ so, daß $S \rightarrow a_1; \dots; a_k \rightarrow U$ ist, und der in S gilt. Die a_i haben alle eine geringere Komplexität als b^* , k -maliges Anwenden der Voraussetzung der Induktion über den Programmaufbau liefert die Existenz eines Knotens $V \geq^* W$ mit $V=V'; \neg [b^*]P^R$, der in U gilt. Nun ist $U \rightarrow b^* \rightarrow T$, Anwenden der Voraussetzung der Induktion über n liefert die Behauptung. \square

Lemma 8. Sei \mathcal{Y} Tableau und sei $Y \in \mathcal{Y}$ freier erfüllbarer normaler Knoten. Ist Y nicht Endknoten, so gibt es einen freien erfüllbaren normalen Knoten $Z \succ Y$.

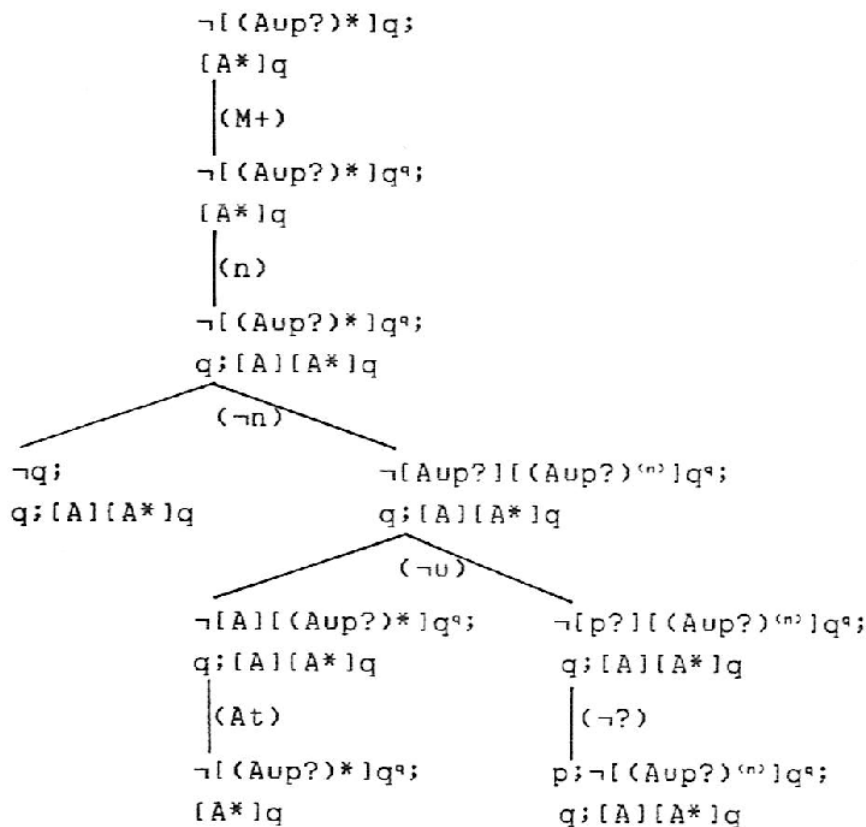
BEWEIS. Wird auf Y nicht $(M+)$ angewandt, ist das die Aussage von Lemma 4. Wird auf Y $(M+)$ angewandt, so ist der Nachfolger von Z ein normaler Knoten von der Gestalt $Y'; \neg[a_1] \dots [a_n] P^p$. Mit Lemma 7 findet man einen $\langle \cdot \rangle$ -Pfad erfüllbarer Knoten zu einem normalen freien Knoten Z , der von der Gestalt $Z'; \neg P^p \dots P^p$ ist, oder der durch Anwendung von $(M-)$ entsteht. In beiden Fällen ist Z erfüllbarer normaler freier Knoten mit $Z \succ Y$. \square

1.7 Der PDL-Kalkül

Definition. Ein Tableau \mathcal{T} heißt *geschlossen*, wenn alle normalen freien Endknoten von \mathcal{T} geschlossen sind.

Im verbleibenden Teil von Abschnitt 1 zeigen wir, daß der hiermit definierte PDL-Kalkül vollständig ist; das heißt, daß jede Formelmengens $X \subseteq \mathcal{F}$ genau dann erfüllbar ist, wenn kein geschlossenes Tableau für X existiert.

Doch zunächst geben wir ein Beispiel eines geschlossenen Tableaus für die Menge $X = \{\neg[(Aup?)*]q, [A^*]q\}$.



Dieses Tableau hat drei Endknoten, deren Merkmale wir kurz betrachten wollen.

- Der Endknoten $\{\neg q; q; [A][A^*]q\}$ ist geschlossen. Er ist normal und frei, und jeder derartige Endknoten eines geschlossenen Tableaus ist geschlossen.

- Der Endknoten $\{ \neg[(A \cup ?)^*]q^a; [A^*]q \}$ ist nicht geschlossen. Er ist jedoch belastet, und er besitzt einen Vorgänger, dem dieselbe Menge zugeordnet ist, wobei außerdem jeder Knoten zwischen ihnen belastet ist. Somit ist er wegen Bedingung 6) Endknoten.
- Der Endknoten $\{ p; \neg[(A \cup ?)^{(\omega)}]q^a; q; [A][A^*]q \}$ ist ein $\neg[a^{(\omega)}]$ -Knoten. Auf ihn darf keine Regel angewandt werden.

Wir bemerken, daß jeder Endknoten eines geschlossenen Tableaus mindestens eines dieser Merkmale aufweist.

Definition. Eine normale Formelmengemenge $X \subseteq \mathcal{F}$ heißt *inkonsistent*, wenn ein geschlossenes Tableau für X existiert, sonst heißt X *konsistent*.

Korrektheitssatz. Jede erfüllbare Formelmengemenge $X \subseteq \mathcal{F}$ ist konsistent.

BEWEIS. Aus Lemma 8 folgt unmittelbar, daß jedes Tableau für eine erfüllbare Formelmengemenge X einen freien erfüllbaren normalen Endknoten hat. Dieser ist insbesondere offen. Damit ist X konsistent. \square

Um zu zeigen, daß umgekehrt jede konsistente Formelmengemenge X auch erfüllbar ist, zeigen wir, wie man aus den offenen Tableaus für X ein Modell für X konstruieren kann. Dazu definieren wir eine bestimmte Klasse von Kripke-Modellen, die Klasse der *Modellgraphen*.

1.8 Modellgraphen

Definition. Eine Formelmengensammlung X heißt *saturiert*, wenn sie folgende Bedingungen erfüllt:

- $\neg\neg P \in X \Rightarrow P \in X$
- $P \wedge Q \in X \Rightarrow P \in X \text{ und } Q \in X$
- $\neg(P \wedge Q) \in X \Rightarrow \neg P \in X \text{ oder } \neg Q \in X$
- $[a; b] P \in X \Rightarrow [a] P \in X \text{ und } [b] P \in X$
- $[a \cup b] P \in X \Rightarrow [a] P \in X \text{ und } [b] P \in X$
- $[Q?] P \in X \Rightarrow \neg Q \in X \text{ oder } P \in X$
- $[a^*] P \in X \Rightarrow P \in X \text{ und } [a][a^*] P \in X$
- $\neg[a; b] P \in X \Rightarrow \neg[a] P \in X \text{ oder } \neg[b] P \in X$
- $\neg[a \cup b] P \in X \Rightarrow \neg[a] P \in X \text{ oder } \neg[b] P \in X$
- $\neg[Q?] P \in X \Rightarrow Q \in X \text{ und } \neg P \in X$
- $\neg[a^*] P \in X \Rightarrow \neg P \in X \text{ oder } \neg[a][a^*] P \in X$

Definition. Ein *Modellgraph* ist ein Kripke-Modell $\mu = (g, \pi, \nu)$, das den Bedingungen i) bis iv) genügt.

- i) Jedes Element von g ist eine saturierte Formelmengensammlung X mit $0 \in X$ und $p \in X \Rightarrow \neg p \notin X$.
- ii) Ist $X \in g$ und $p \in X$, so ist $X \in \nu(p)$.
- iii) Ist $(X, Y) \in \pi(A)$ und $[A] P \in X$, so ist $P \in Y$.
- iv) Ist $X \in g$ und $\neg[a] P \in X$, so gibt es $Y \in g$ mit $(X, Y) \in \pi'(a)$ und $\neg P \in Y$.

Dabei ist π' eine Funktion auf \mathcal{P} , die auf \mathcal{P}_0 mit π übereinstimmt, die für Tests durch

$$\pi'(P?) := \{(X, X) \mid X \in g \text{ und } P \in X\}$$

definiert ist und die für Programme der Gestalt $a; b$, $a \cup b$ und a^* wie π induktiv definiert ist.

Offensichtlich läßt sich zu jedem Kripke-Modell $\mu = (g, \pi, \nu)$ ein Modellgraph konstruieren:

Für $S \in g$ sei $X_S := \{P \in \mathcal{P} \mid S \Vdash P\}$, es sei $g' := \{X_S \mid S \in g\}$, weiter sei $\pi'(A) := \{(X_S, X_T) \mid X_S \Vdash X_T\}$ für alle $A \in \mathcal{P}_0$ und $\nu'(p) := \{X_S \mid p \in X_S\}$ für alle $p \in \mathcal{P}_0$.

Man zeigt leicht, daß $\mu' := (g', \pi', \nu')$ die Bedingungen i) bis iv) erfüllt.

Lemma 9. Ist $\mu=(g,\pi,\nu)$ Modellgraph, $X \in g$ und $P \in X$, so gilt $X \Vdash P$.

BEWEIS. Wir zeigen durch simultane Induktion über den Formel- und Programmaufbau, daß

(+) $P \in X \Rightarrow X \Vdash P$ (mit $\Vdash = \Vdash^*$)

(-) $\neg P \in X \Rightarrow X \nVdash P$

für jede Formel P und

(o) $[a]P \in X \Rightarrow$ ist $(X,Y) \in \pi(a)$, so ist $P \in Y$

für jedes Programm a (und jede Formel P) gilt.

Ist $P=0$ oder ist P Aussagenvariable, so gelten (+) und (-) auf Grund der Bedingungen i) und ii).

Sei also jetzt P eine Formel, für deren Subformeln (+) und (-) bereits bewiesen sind. Außerdem sei (o) für alle Programme bewiesen, die in P auftauchen.

Sei $P=\neg Q$. Ist $P \in X$, so gilt nach (-) und Induktionsvoraussetzung $X \nVdash Q$, also $X \Vdash \neg Q=P$, und damit gilt (+).

Ist $\neg P=\neg\neg Q \in X$, so ist wegen der Saturiertheit von X auch $Q \in X$ und $X \Vdash Q$ nach Induktionsvoraussetzung. Damit gilt aber $X \nVdash \neg Q=P$, und das ist (-).

Sei $P=Q \wedge R$. Ist $P \in X$, so sind $Q \in X$ und $R \in X$, was nach Induktionsvoraussetzung $X \Vdash Q$ und $X \Vdash R$, somit $X \Vdash Q \wedge R=P$ bedeutet, somit gilt (+). Ist $\neg P \in X$, so ist $\neg Q \in X$ oder $\neg R \in X$. Somit gilt nicht $X \Vdash Q$ und $X \Vdash R$, das heißt $X \nVdash Q \wedge R=P$, und das ist (-).

Sei $P=[a]Q$. Ist $P \in X$, so folgt nach Voraussetzung der Induktion über a , daß für alle Y mit $(X,Y) \in \pi(a)$ schon $Q \in Y$ ist. Nach Voraussetzung der Induktion über P folgt aus $Q \in Y$ schon $Y \Vdash Q$, somit gilt $X \Vdash [a]Q$. Ist $\neg P \in X$, so gilt für jeden Test R , der in a vorkommt, bereits (+). Aus $(X,Y) \in \pi'(R?)$ folgt also $(X,Y) \in \pi(R?)$, ist somit $(X,Y) \in \pi'(a)$, so ist auch $(X,Y) \in \pi(a)$. Wegen iv) existiert also ein Y mit $(X,Y) \in \pi(a)$ und $\neg Q \in Y$. Nach Induktionsvoraussetzung gilt $Y \nVdash Q$, somit gilt $X \nVdash [a]Q$.

Für atomare Programme gilt (o) stets wegen der Bedingung iii). Sei also $[a]P \in X$ und a ein nichtatomares Programm, für dessen Subprogramme (o) bereits bewiesen ist. Außerdem sei (-) bereits bewiesen für alle Tests, die

in a auftauchen.

Sei $a=R?$. Wegen der Saturiertheit ist $\neg REX$ oder PEX . Ist $(X,Y) \in \pi(R?)$, so ist $Y=X$ und $X \Vdash R$. Wegen $(-)$ ist dann aber $\neg R \notin X$, also ist $PEX=Y$.

Sei $a=buc$. Wegen der Saturiertheit ist $[b]PEX$ und $[c]PEX$. Ist nun $(X,Y) \in \pi(a)$, so ist $(X,Y) \in \pi(b)$ oder $(X,Y) \in \pi(c)$. In jedem Fall ist nach Induktionsvoraussetzung PEY .

Sei $a=b;c$. Es ist $[b][c]PEX$. Ist $(X,Y) \in \pi(a)$, so gibt es $Z \in g$ mit $(X,Z) \in \pi(b)$ und $(Z,Y) \in \pi(c)$. Nach Induktionsvoraussetzung ist $[c]PEZ$ und PEY .

Sei $a=b^*$. Ist $(X,Y) \in \pi(b^*)$, so gibt es ein minimales $n \in \omega$ mit $(X,Y) \in \pi(b^n)$. Wir beweisen (o) nun durch Induktion über n . Der Fall $n=0$ ist klar, denn dann ist $X=Y$, und wegen der Saturiertheit ist schon PEX . Sei also die Behauptung bewiesen für $m=n-1$. Es ist $[b][b^*]PEX$, und es gibt ein $Z \in g$ mit $(X,Z) \in \pi(b)$ und $(Z,Y) \in \pi(b^{n-1})$. Nach der Voraussetzung der Induktion über a folgt $[b^*]PEZ$, und mit der Voraussetzung der Induktion über n folgt PEY . \square

1.9 Konsistente Knoten

Um die Konstruktion von Modellgraphen aus Tableaus vornehmen zu können, definieren wir für jedes Tableau folgende Relationen.

Definition. Seien $X < Y$ Knoten eines Tableaus. Wir schreiben

- $X \rightarrow Y$, wenn auf genau einen Knoten Z mit $X < Z < Y$ die kritische Regel angewandt wird, und zwar auf eine Formel der Gestalt $\neg(A)P^*$.
- $X \rightarrow? Y$, wenn der Pfad von X zu Y unkritisch ist und ein Z mit $X < Z < Y$ und $P \in Z$ existiert.
- $X \rightarrow a \rightarrow b \rightarrow Y$, wenn $X \rightarrow a \rightarrow Y$ oder $X \rightarrow b \rightarrow Y$.
- $X \rightarrow a ; b \rightarrow Y$, wenn ein Knoten Z mit $X < Z < Y$ existiert so, daß $X \rightarrow a \rightarrow Z$ und $Z \rightarrow b \rightarrow Y$.
- $X \rightarrow a^* \rightarrow Y$, wenn der Pfad von X zu Y unkritisch ist, oder wenn $X \rightarrow a^i \rightarrow Y$ für ein $i \geq 1$ ist.

Sind ferner $X < Y$ Knoten so, daß Y frei ist und kein freier Knoten Z mit $X < Z < Y$ existiert, so nennen wir Y einen *ersten freien Nachfolger* von X .

Weiter benötigen wir noch einige Ergebnisse über das Auftreten von konsistenten Knoten in Tableaus.

Lemma 10. Jeder Knoten $X = X'; \neg(a_1) \dots (a_n)P^*$, der durch Anwendung von $(M+)$ entsteht, hat einen freien normalen Nachfolger Y . Ist Y ein erster freier Nachfolger eines Knotens $X = X'; \neg(a_1) \dots (a_n)P^*$, und entsteht Y nicht durch Anwendung von $(M-)$, so ist Y von der Gestalt $Y = Y'; \neg P$, und es gilt $X \rightarrow a_1 ; \dots ; a_n \rightarrow Y$.

BEWEIS. Durch Induktion über n und den Aufbau der a_i . \square

Damit hat insbesondere jeder freie normale Knoten eines Tableaus, der nicht schon Endknoten ist, mindestens einen freien normalen Nachfolger.

Lemma 11. Sei $X \in J$ freier normaler Knoten mit den ersten freien normalen Nachfolgern Y_1, \dots, Y_n ($n \geq 1$). Sind alle Y_i inkonsistent, so ist X inkonsistent.

BEWEIS. Für jedes i sei J_i ein geschlossenes Tableau für Y_i .

Ist X Knoten eines Tableaus J_i , so ist dasjenige Teilttableau von J_i , welches aus allen Knoten $Y \in J_i$ mit $X \leq Y$ besteht, ein geschlossenes Tableau für X .

Ist X in keinem J_i Knoten, so bildet man ein neues Tableau für X , in dem man an den Knoten Y_i von J das alte Tableau "abschneidet" und die jeweiligen J_i "anheftet". Dieses neue Tableau ist ein geschlossenes PDL-Tableau für X .

In beiden Fällen folgt, daß X inkonsistent ist. \square

Definition. Eine belastete normale Formelmengung X nennen wir *inkonsistent*, wenn $X^{-1} = \{P \mid PEX \text{ oder } P^*EX \text{ für gewisses } R\}$ inkonsistent ist, sonst *konsistent*.

Lemma 12. Ist X freier normaler konsistenter Knoten in einem Tableau J , so gibt es einen ersten freien normalen konsistenten Nachfolger $Y \succ X$ so, daß jeder normale Knoten auf dem Pfad von X zu Y konsistent ist.

BEWEIS. Seien Y_1, \dots, Y_n die ersten freien normalen konsistenten Nachfolger von X (Lemma 11 garantiert die Existenz solcher Knoten). Nehmen wir an, es gibt für jedes i einen normalen inkonsistenten Knoten Z_i mit $X < Z_i < Y_i$. Diese Knoten sind alle belastet. Man erhält dann durch Anwenden von (M-) auf die Z_i ein Tableau, in dem sämtliche ersten freien normalen Nachfolger von X inkonsistent sind. Nach Lemma 11 ist dann X inkonsistent; die Annahme widerspricht also der Voraussetzung. \square

1.10 Der Vollständigkeitssatz

Definition. Wir nennen ein Tableau \mathcal{T} *saturierend*, wenn die kritische Regel nur auf einfache Knoten angewandt wird.

Modellexistenzsatz. Ist Z_0 eine normale konsistente Formelmengende, so gibt es einen Modellgraphen $\mu=(g, \nu, \pi)$ und ein $S \in g$ mit $Z_0 \subseteq S$.

BEWEIS. Sei \mathcal{T}_0 ein maximales lokales Tableau für Z_0 , in dem nicht (M+) angewandt wird. Weiter sei M_0 die kleinste Menge von Tableaus, die

- a) \mathcal{T}_0 enthält und
- b) für jeden einfachen konsistenten Knoten Y eines $\mathcal{T} \in M_0$ jedes maximale saturierende Tableau für Y enthält.

Für Knoten $X, Y \in \mathcal{T} \in M_0$ sei

$$S_{X,Y} := \{f(P) \mid \text{es gibt } Z \text{ mit } X \leq Z \leq Y \text{ und } P \in Z\},$$

$$g := \{S_{X,Y} \mid X \text{ und } Y \text{ sind konsistente Wurzel und Endknoten eines lokalen Teilttableaus eines } \mathcal{T} \in M_0, \text{ und } Y \text{ ist einfach}\}$$

Jedes $S_{X,Y} \in g$ ist eine saturierte Formelmengende und erfüllt Bedingung i) für Modellgraphen. Wir bemerken, daß $|g| \leq 2^{|\mathcal{P}_0|}$ ist. Weiter sei für alle $S, T \in g$

$$S \in \nu(p) \iff p \in S \quad \text{für alle } p \in \mathcal{P}_0,$$

$$(S, T) \in \pi(A) \iff S_A \subseteq T \quad \text{für alle } A \in \mathcal{P}_0.$$

Mit diesen Definitionen genügt $\mu=(g, \nu, \pi)$ den Bedingungen ii) und iii) für Modellgraphen. Durch Induktion über a zeigt man leicht, daß gilt:

Sind $X, Y, X', Y' \in \mathcal{T} \in M_0$ mit $S_{X,Y} \in g$ und $S_{X',Y'} \in g$, sind ferner Z, Z' Knoten mit $X \leq Z \leq Y$, $X' \leq Z' \leq Y'$ und $Z \xrightarrow{a} Z'$, so ist $(S_{X,Y}, S_{X',Y'}) \in \pi'(a)$.

Wir zeigen nun, daß μ auch Bedingung iv) erfüllt. Sei dazu $S \in g$ und $\neg(a) \in P \in S$. Es ist $S = S_{X,Y}$ für gewisse Knoten X, Y eines $\mathcal{T} \in M_0$. Sei $X = X_1 \triangleleft \dots \triangleleft X_n = Y$ der Pfad von X zu Y . Es gibt ein X_j mit $\neg(a) \in P \in X_j$ (eventuell liegt $\neg(a) \in P$ als n -Formel vor; diesen Umstand ignorieren wir

jetzt). O.B.d.A. entstehe kein $X_i \triangleright X_j$ durch Anwendung von (M-) oder (M+).

Wir bilden nun den Pfad $Y_1 \triangleleft \dots \triangleleft Y_n$. Es entstehe Y_j aus X_j^- durch Markierung von $\neg[a]P$ mit P , und für $k > j$ entstehe Y_k aus Y_{k-1} wie X_k^- aus X_{k-1}^- (damit gilt stets $Y_k^- = X_k^-$). Es können zwei Fälle eintreten:

- 1) Ein Y_k , $k > j$ ist frei. Dann ist $Y_j \neg a \triangleright Y_k$, insbesondere ist $(S, S) \in \pi'(a)$ und $\neg PES$.
- 2) Alle Y_k , $k > j$ sind belastet. Der Knoten Y_n enthält dann eine Formel von der Gestalt $\neg[A][a_1] \dots [a_1]P^n$. Nun gibt es aber ein Tableau $\mathcal{J}' \in \mathcal{M}_0$, das die Wurzel X_n^- hat, deren Nachfolger Y_n ist, und in dem nicht (M-) angewandt wird. Nach Lemma 12 hat Y_n in diesem Tableau einen ersten freien normalen konsistenten Nachfolger Z so, daß jeder normale Knoten auf dem Pfad von Y_n zu Z konsistent ist. Außerdem existiert in \mathcal{J}' ein einfacher konsistenter Nachfolger Z' von Z , der im selben lokalen Teiltabelleau von \mathcal{J}' wie Z ist, denn \mathcal{J}' ist maximal.

Betrachten wir nun den Pfad

$$X_1 \triangleleft \dots \triangleleft X_j \triangleleft Y_j \triangleleft \dots \triangleleft Y_n \triangleleft \dots \triangleleft Z \triangleleft \dots \triangleleft Z'$$

Jeder normale Knoten des Pfades ist konsistent. Jeder maximale unkritische Teilpfad dieses Pfades endet mit einem einfachen Knoten, da \mathcal{J}' saturierend ist. Zu allen Knoten V, W , die Wurzel und Endknoten eines unkritischen Teilpfades sind, existiert ein $S_{V,W} \in \mathcal{E}_g$. Insbesondere existiert ein $S' \in \mathcal{E}_g$ mit $Z \in S'$ und demzufolge $\neg PES'$. Nach Lemma 10 ist $Y_j \neg a \triangleright Z$, somit ist $(S, S') \in \pi'(a)$.

Damit erfüllt $\mu = (g, \nu, \pi)$ auch Bedingung iv). \square

Vollständigkeitssatz. Jede Formelmenge $X \in \mathcal{F}$ ist genau dann konsistent, wenn sie erfüllbar ist.

BEWEIS. Aus dem Modellexistenzsatz und Lemma 9 folgt, daß jede konsistente Menge erfüllbar ist. Mit dem Korrektheitssatz folgt die Behauptung. \square

1.11 Erweiterte PDL-Tableaus

Häufig kann man einen vollständigen Tableau-Kalkül durch Hinzunahme weiterer Regeln so erweitern, daß die Konstruktion geschlossener Tableaus für inkonsistente Mengen vereinfacht wird. Diesen Umstand werden wir uns im Abschnitt 2 zunutze machen. Dort werden wir (zusätzlich) die folgenden Regeln benutzen:

$$(*) \frac{X;[a^*;b]P}{X;[b]P \mid X;[a][a^*;b]P} \quad (?;) \frac{X;[(1?;a]P}{X;[a]P}$$

$$(F-) \frac{X;Y}{X} \quad Y \text{ frei} \quad (1?) \frac{X;[(1?]P}{X;P}$$

Außerdem soll jede Anwendung einer Regel auf eine normale freie Formel wahlweise als Austausch (also wie bisher) oder als Hinzufügung von Formeln verstanden werden können. Ein Knoten $X;P \wedge Q$ kann so durch Anwendung von (\wedge) den Nachfolger $X;P \wedge Q;P;Q$ erhalten.

Definition. Ein *erweitertes* PDL-Tableau ist ein Tableau, das mit den bisher definierten Regeln konstruiert ist. Ein solches Tableau \mathcal{T} heißt genau dann geschlossen, wenn alle normalen freien Endknoten geschlossen sind.

Man überzeugt sich schnell davon, daß für eine erfüllbare Menge X kein geschlossenes erweitertes PDL-Tableau existiert. Es folgt dann unmittelbar, daß eine Menge X sogar genau dann erfüllbar ist, wenn kein geschlossenes erweitertes PDL-Tableau für sie existiert. Durch Angabe eines geschlossenen erweiterten PDL-Tableaus für eine Menge X wird diese also auch als nicht erfüllbar nachgewiesen.

2 Interpolation in PDL

2.1 Der Interpolationssatz

Viele Modallogiken, darunter alle im Abschnitt 1.2 erwähnten, besitzen eine Interpolationseigenschaft. Der Nachweis dafür kann mit geeigneten vollständigen Tableau-Kalkülen erbracht werden (RAUTENBERG [12]). In diesem Abschnitt werden wir eine Interpolationseigenschaft für PDL formulieren und mit Hilfe des PDL-Kalküls nachweisen, daß PDL diese Eigenschaft tatsächlich besitzt.

Zu diesem Zweck bezeichnen wir mit $\mathcal{F}_0(P)$ bzw. $\mathcal{P}_0(P)$ die Mengen der Aussagenvariablen bzw. der atomaren Programme, die in P auftauchen. Ferner sei $S(P) := \mathcal{F}_0(P) \cup \mathcal{P}_0(P)$. Analog sei $S(a)$ für alle Programme a definiert. Schließlich sei für eine Formelmengens X definiert $S(X) := \bigcup_{P \in X} S(P)$.

Definition. Ein *Interpoland* für das Paar (X_1, X_2) von Formelmengen ist eine Formel R mit $S(R) \subseteq S(X_1) \cap S(X_2)$, für die $X_1; \neg R$ und $X_2; R$ inkonsistent sind.

Interpolationssatz. Seien P, Q Formeln mit $\models P \rightarrow Q$. Dann gibt es eine Formel R mit $S(R) \subseteq S(P) \cap S(Q)$, für die $\models P \rightarrow R$ und $\models R \rightarrow Q$ gilt.

Eine Formel R mit dieser Eigenschaft nennen wir (etwas ungenau) auch einen Interpolanden für P und Q . Tatsächlich handelt es sich (nach unserer Definition) um einen Interpolanden für das Paar $(\{P\}, \{\neg Q\})$. Wir beweisen im folgenden, daß für jedes Paar (X_1, X_2) ein Interpoland existiert, sofern $X_1; X_2$ inkonsistent ist. Der Interpolationssatz ergibt sich dann daraus mit $X_1 = \{P\}$ und $X_2 = \{\neg Q\}$, denn die Inkonsistenz von $P; \neg Q$ bedeutet ja $\models P \rightarrow Q$.

2.2 Tableaus für Paare von Formelmengen

Wir wollen nun den Begriff des Tableaus für ein Paar X_1/X_2 von Formelmengen einführen. Es handelt sich dabei wieder um einen endlichen Baum (T, \triangleleft) ; jedem Knoten $t \in T$ wird aber durch zwei Funktionen x_1 und x_2 ein Paar von Formelmengen $x_1(t)/x_2(t)$ zugeordnet. Wir bezeichnen oft auch einen Knoten mit dem ihm zugeordneten Paar. Ein Knoten Y_1/Y_2 heißt frei, belastet, geschlossen, normal, n-Knoten oder $\neg[a^{(n)}]$ -Knoten, wenn $Y_1; Y_2$ so heißt. Ein solches Tableau wird ausgehend von der Wurzel X_1/X_2 durch Anwendung der Regeln des PDL-Kalküls konstruiert. Wir müssen aber erklären, wie die Anwendung einer Regel auf eine Formel in einer Komponente eines Paares Y_1/Y_2 zu verstehen ist.

- Entstehen bei Anwendung einer beliebigen Regel (nicht (At)) auf eine Formel PEY_1 (Y_1 Knoten eines PDL-Tableaus) die Formelmengen Z_1, \dots, Z_n ($n=1$ oder $n=2$), so entstehen bei Anwendung derselben Regel auf PEY_1 (Y_1 Komponente des Knotens Y_1/Y_2) die Paare $Z_1/Y_2, \dots, Z_n/Y_2$. Analog bei Anwendung auf eine Formel PEY_2 .
- Bei Anwendung von (At) auf eine Formel $\neg[A]P^R EY_1$ (Y_1 Komponente des Knotens Y_1/Y_2 und $Y_1=Y; \neg[A]P^R$ für gewisses Y) entsteht das Paar $Y_A; \neg P^{R^*} / (Y_2)_A$. Analog bei Anwendung auf eine Formel $\neg[A]P^R EY_2$.

Bei dieser Konstruktion sind ebenfalls die Bedingungen 1) bis 5) und 7) zu beachten. Die Bedingung 6) muß jedoch für Tableaus für Paare anders formuliert werden.

- 6') Ein normaler Knoten $x_1(t)/x_2(t)$, zu dem ein $s < t$ mit $x_1(s)=x_1(t)$ und $x_2(s)=x_2(t)$ existiert, ist Endknoten, wenn der Pfad von s zu t kritisch ist, und wenn der Pfad zudem belastet ist, wenn $x_1(t)/x_2(t)$ belastet ist.

Zusätzlich ist bei der Konstruktion eines Tableaus für

ein Paar stets folgende Vorschrift zu beachten:

- 8) Existieren zwei Knoten s und t mit der selben belasteten Komponente in einem belasteten Teiltabelleau, und entstehen deren jeweilige Nachfolger durch Anwendung einer Regel (nicht (M-)) auf die belastete Komponente, so wird auf beide Knoten dieselbe Regel auf dieselbe Formel angewandt.

Definition. Wir nennen ein Tableau für ein Paar X_1/X_2 geschlossen, wenn alle normalen freien Endknoten geschlossen sind. Weiter nennen wir ein normales Paar X_1/X_2 inkonsistent, wenn ein geschlossenes Tableau für X_1/X_2 existiert, sonst nennen wir es konsistent.

Lemma 13. Ein normales Paar X_1/X_2 ist genau dann konsistent, wenn $X_1;X_2$ konsistent ist.

BEWEIS. Ist X_1/X_2 inkonsistent, so gibt es ein geschlossenes Tableau für X_1/X_2 . Man erhält durch Ersetzen jedes Knotens Y_1/Y_2 durch $Y_1;Y_2$ im wesentlichen ein erweitertes PDL-Tableau. Es kann lediglich passieren, daß Knoten $s < t$ existieren, die Bedingung 6) verletzen. Ist t belastet, so kann man das Tableau dort einfach "abschneiden"; sonst kann man durch "Herausschneiden" des Pfades von s zu t ein erweitertes PDL-Tableau bilden, wie es in 1.11 definiert wurde. Dieses Tableau ist ebenfalls geschlossen, somit ist $X_1;X_2$ inkonsistent.

Ist andererseits X_1/X_2 konsistent, so ist jedes Tableau für X_1/X_2 nicht geschlossen. Man kann dann (wieder nach Ersetzen jedes Knotens Y_1/Y_2 durch $Y_1;Y_2$) aus diesen Tableaus genau wie im Modellexistenzsatz beschrieben einen Modellgraphen konstruieren, in dem ein Zustand die Menge $X_1;X_2$ umfaßt. In diesem Fall ist also auch $X_1;X_2$ konsistent. \square

Korollar. Ist $X_1;X_2$ inkonsistent, so gibt es ein geschlossenes Tableau für X_1/X_2 .

2.3 Konstruktion des Interpolanden

In diesem Abschnitt legen wir dar, wie man für X_1/X_2 einen Interpoland konstruiert, wenn $X_1;X_2$ inkonsistent ist. Man benötigt dazu ein geschlossenes Tableau \mathcal{J} für X_1/X_2 . Dieses Tableau wird zunächst vereinfacht.

Definition. Ist \mathcal{J} ein Tableau für das Paar X_1/X_2 , so entsteht das Tableau \mathcal{J}' aus \mathcal{J} durch "herausschneiden" aller n -Knoten. Sind also $s \triangleleft t$ zwei normale Knoten in \mathcal{J} , zwischen denen sich ausschließlich n -Knoten befinden, so gilt $s \triangleleft t$ in \mathcal{J}' .

Wir bemerken, daß sich $x_1(s)/x_2(s)$ von $x_1(t)/x_2(t)$ nur in einer Komponente unterscheidet. Denn wird z.B. auf eine Formel in $x_1(s)$ die Regel (n) oder $(\neg n)$ angewandt, so wird (nach Bedingung 3)) auf jeden Knoten u mit $s \triangleleft u \triangleleft t$ eine Regel auf $x_1(u)$ angewandt, da nur diese Menge eine n -Formel enthält.

Da \mathcal{J}' keine n -Knoten mehr enthält, existiert auch kein Endknoten von \mathcal{J}' der ein $\neg[a^{(n)}]$ -Knoten wäre. Jeder Endknoten ist also entweder geschlossen oder besitzt einen Vorgänger, dem das gleiche Paar zugeordnet ist. Besitzt \mathcal{J}' sogar nur geschlossene Endknoten, so kann man einen Interpolanden für X_1/X_2 direkt aus \mathcal{J}' konstruieren. Dies geschieht dadurch, daß man zunächst für alle Endknoten von \mathcal{J}' einen Interpolanden angibt.

Im folgenden werden wir Formeln, die Interpolanden für gewisse Paare sind, mit I, I_1, I_2, \dots bezeichnen.

Lemma 14. Zu jedem geschlossenen Knoten Y_1/Y_2 von \mathcal{J}' existiert ein Interpoland.

BEWEIS. Das Paar Y_1/Y_2 ist geschlossen, wenn $Y_1;Y_2$ geschlossen ist. Es liegt also wenigstens einer der folgenden Fälle vor:

- $0 \in Y_1$ oder $P; \neg P \in Y_1$ für eine Formel P :
dann gilt $Y_1 \models 0$ und $Y_2 \models \neg 0$, sei also $I := 0$.
- $0 \in Y_2$ oder $P; \neg P \in Y_2$ für eine Formel P :
dann gilt $Y_1 \models \neg 0$ und $Y_2 \models 0$, sei also $I := 0$.
- $P \in Y_1$ und $\neg P \in Y_2$ für eine Formel P :
dann gilt $Y_1 \models P$ und $Y_2 \models \neg P$, sei also $I := P$.
- $\neg P \in Y_1$ und $P \in Y_2$ für eine Formel P :
dann gilt $Y_1 \models \neg P$ und $Y_2 \models P$, sei also $I := \neg P$.

In jedem der Fälle gilt zudem $S(I) \subseteq S(Y_1) \cap S(Y_2)$, also ist I in jedem Fall Interpoland von Y_1/Y_2 . \square

Anschließend kann man für jeden Knoten, für dessen Nachfolger bereits Interpolanden gefunden sind, ebenfalls einen Interpoland angeben. Man konstruiert also den Interpolanden von X_1/X_2 "von den Blättern zur Wurzel hin".

Lemma 15. Sei Y_1/Y_2 ein Knoten von \mathcal{T}^1 , für dessen n direkte Nachfolger bereits Interpolanden I_1, \dots, I_n gefunden sind. Dann existiert ein Interpoland für Y_1/Y_2 .

BEWEIS. Wenn die n Nachfolger durch Anwendung irgend einer Regel außer (At) entstanden sind, gilt einer der folgenden Fälle:

- Die Nachfolger unterscheiden sich von Y_1/Y_2 nur in der zweiten Komponente, sie sind von der Gestalt $Y_1/Z_1, \dots, Y_1/Z_n$.
Offensichtlich gilt $Y_1 \models I_1 \wedge \dots \wedge I_n$. Weiter gilt Y_2 nur in einem Zustand, in dem auch ein Z_i gilt; das heißt aber, wegen $Z_i \models \neg I_i$ für jedes $i \leq n$, daß $Y_2 \models \neg I_1 \vee \dots \vee \neg I_n$, somit $Y_2 \models \neg (I_1 \wedge \dots \wedge I_n)$. Sei also $I := I_1 \wedge \dots \wedge I_n$. Da $S(Z_i) \subseteq S(Y_2)$ für jedes i ist, folgt $S(I) \subseteq S(Y_1) \cap S(Y_2)$; I ist somit Interpoland für Y_1/Y_2 .
- Die Nachfolger unterscheiden sich von Y_1/Y_2 nur in der ersten Komponente, sie sind von der Gestalt $Z_1/Y_2, \dots, Z_n/Y_2$.
Offensichtlich gilt $Y_2 \models \neg I_1 \wedge \dots \wedge \neg I_n$, somit $Y_2 \models \neg (I_1 \vee \dots \vee I_n)$. Weiter gilt Y_1 nur in einem Zustand, in dem auch ein Z_i gilt; das heißt aber, wegen $Z_i \models I_i$

für jedes $i \leq n$, daß $Y_i \models I_i \vee \dots \vee I_n$. Sei also $I := I_1 \vee \dots \vee I_n$. Da $S(Z_i) \subseteq S(Y_2)$ für jedes i ist, folgt $S(I) \subseteq S(Y_1) \cap S(Y_2)$; I ist somit Interpoland für Y_1/Y_2 .

Bevor wir den zweiten Fall behandeln, überlegen wir, daß aus $Y_A \models P$ stets $Y \models [A]P$ folgt. Denn gilt $S \Vdash Y$ für irgend ein (g, ν, π) und ein $S \in g$, so gilt $T \Vdash Y_A$ und somit $T \Vdash P$ für alle $T \in g$ mit $S \rightarrow A \rightarrow T$. Das heißt aber $S \Vdash [A]P$. Weiter folgt aus $Y_A; \neg R \models P$ stets $Y; \neg[A]R \models \neg[A]\neg P$, denn gilt $S \Vdash Y; \neg[A]R$ für irgend ein (g, ν, π) und ein $S \in g$, so existiert ein $T \in g$ mit $S \rightarrow A \rightarrow T$ und $T \Vdash Y_A; \neg R$. Da dann $T \Vdash P$ gilt, heißt das aber $S \Vdash \neg[A]\neg P$.

Wenn nun Y_1/Y_2 einen Nachfolger Z_1/Z_2 hat, der durch Anwendung der kritischen Regel entstanden ist, und für den ein Interpoland I bereits gefunden ist, gilt einer der folgenden Fälle:

- Y_1 enthält eine markierte Formel der Gestalt $\neg[A]P^*$. Dann ist $Z_1 = \neg P^{* \wedge P}; (Y_1)_A$ und $Z_2 = (Y_2)_A$. Ist Z_2 nichtleer, so ist $A \in S(Y_1) \cap S(Y_2)$. Wie wir gesehen haben, folgt $Y_1 \models \neg[A]\neg I$ aus $\neg P^{* \wedge P}; (Y_1)_A \models I$ und $Y_2 \models [A]\neg I$ aus $(Y_2)_A \models \neg I$. Damit ist $\neg[A]\neg I$ Interpoland für Y_1/Y_2 . Ist aber Z_2 leer, so ist Y_1 schon inkonsistent. Dann gilt $Y_1 \models 0$ und $Y_2 \models \neg 0$, und 0 ist Interpoland für Y_1/Y_2 .
- Y_2 enthält eine markierte Formel der Gestalt $\neg[A]P^*$. Dann ist $Z_1 = (Y_1)_A$ und $Z_2 = \neg P^{* \wedge P}; (Y_2)_A$. Ist Z_1 nichtleer, so ist $A \in S(Y_1) \cap S(Y_2)$. Wie wir gesehen haben, folgt $Y_1 \models I$ aus $(Y_1)_A \models I$ und $Y_2 \models \neg[A]I$ aus $\neg P^{* \wedge P}; (Y_2)_A \models \neg I$. Damit ist $[A]I$ Interpoland für Y_1/Y_2 . Ist aber Z_1 leer, so ist Y_2 schon inkonsistent. Dann gilt $Y_1 \models \neg 0$ und $Y_2 \models 0$, und $\neg 0$ ist Interpoland für Y_1/Y_2 .

Damit ist das Lemma bewiesen. \square

Im allgemeinen weist \mathcal{J}^i jedoch nicht nur geschlossene Endknoten auf. Es sind ja auch solche belasteten Knoten Endknoten, für die ein Vorgänger im selben belasteten Teiltabelleau von \mathcal{J}^i existiert, dem das gleiche Paar zugeordnet ist.

Das bedeutet, daß es für die Wurzel eines belasteten Teilttableaus im allgemeinen nicht möglich ist, mit den in den Lemmata 14 und 15 beschriebenen Verfahren einen Interpolanden zu konstruieren, da es Nachfolger dieser Wurzel geben kann, die nicht geschlossene Endknoten sind. Wir legen im folgenden dar, daß man jedoch stets einen Interpolanden für eine solche Wurzel konstruieren kann, sofern nur für ihre ersten freien Nachfolger Interpolanden bereits bekannt sind.

Sei also Y_1/Y_2 ein Knoten von \mathcal{J}' , der durch Anwendung von $(M+)$ entsteht; Y_1/Y_2 ist dann die Wurzel eines belasteten Teilttableaus von \mathcal{J}' . Für alle freien Nachfolger von Y_1/Y_2 sei ein Interpoland bereits bekannt. Wir können uns auf den Fall beschränken, daß Y_2 die markierte Formel enthält. Denn ist das nicht der Fall, können wir in allen Knoten des Tableaus die erste und zweite Komponente vertauschen (ist I Interpoland für Y_1/Y_2 , so ist $-I$ Interpoland für Y_2/Y_1 ; die bisher gefundenen Interpolanden sind also nur durch ihre Negationen zu ersetzen). Des weiteren können wir uns auf den Fall $Y_1 \neq \emptyset$ beschränken, denn ist $Y_1 = \emptyset$, so ist Y_2 schon inkonsistent und -0 ist Interpoland für Y_1 und Y_2 . Wir werden nun angeben, welchen Teil von \mathcal{J}' wir für die Konstruktion des Interpolanden benötigen.

Definition. \mathcal{J}' sei ein Teilttableau von \mathcal{J}' mit der Wurzel Y_1/Y_2 . Ein Knoten $Z_1/Z_2 \succ Y_1/Y_2$ sei genau dann Endknoten von \mathcal{J}' , wenn er minimal ist so, daß er einer der folgenden Bedingungen genügt:

- Auf Z_1/Z_2 wird $(M-)$ angewandt.
- Z_1/Z_2 ist frei.
- Die erste Komponente seines Nachfolgers ist leer.
- Es existiert ein Vorgänger von Z_1/Z_2 , dem das gleiche Paar zugeordnet ist.

Besitzt ein Knoten einen Vorgänger, dem das gleiche Paar zugeordnet ist, werden wir diesen im folgenden auch kurz einen *Vorgänger mit dem gleichen Paar* nennen.

Lemma 16. \mathcal{J}^j hat folgende Eigenschaften:

- a) Jede Komponente jedes Knotens ist nichtleer.
- b) Für jeden Endknoten t , für den kein Vorgänger mit dem gleichen Paar existiert, ist ein Interpoland $I(t)$ bekannt.
- c) Es gibt mindestens einen Endknoten, für den ein Interpoland bekannt ist.

BEWEIS. Die Gültigkeit von a) ist offensichtlich. Sei t ein Endknoten von \mathcal{J}^j , für den kein Vorgänger mit dem gleichen Paar existiert. Ist $x_2(t)$ frei, dann ist schon ein Interpoland für t bekannt. Ist $x_2(t)$ nicht frei, so wird auf t (M-) angewandt, oder die erste Komponente seines Nachfolgers t' in \mathcal{J}^j ist leer (was nur durch Anwendung von (At) auf t geschehen kann). Im ersten Fall ist t' frei, womit für t' und damit auch für t ein Interpoland bereits bekannt ist. Im zweiten Fall ist aber $x_2(t)$ schon inkonsistent, und $\neg 0$ ist somit Interpoland für t . Also gilt auch b).

In \mathcal{J}^j besitzt Y_1/Y_2 mindestens einen ersten freien Nachfolger t . Für diesen Knoten ist ein Interpoland bekannt. Entweder ist dieser Knoten auch in \mathcal{J}^j enthalten, oder aber es gibt einen Knoten $s(t)$, der Endknoten von \mathcal{J}^j ist. Für s aber existiert kein Vorgänger mit dem gleichen Paar in \mathcal{J}^j . Nach b) ist somit ein Interpoland für s bekannt. Somit gilt auch c). \square

Definition. Für jeden Knoten t von \mathcal{J}^j sei $K(t)$ die Konjunktion über alle $P \in x_1(t)$. Für jede Menge T von Knoten von \mathcal{J}^j sei $D(T)$ die Disjunktion über alle $K(t)$ mit $t \in T$. Für $T = \emptyset$ sei $D(T) = 0$. Wir bemerken, daß für $T_1 \subseteq T_2$ aus $D(T_2) \neq 0$ stets $D(T_1) \neq 0$ folgt. Weiter bezeichne $T(Y)$ die Menge aller Knoten t von \mathcal{J}^j mit $x_2(t) = Y$.

Für jedes Y , das als zweite Komponente eines Knotens in \mathcal{J} auftaucht, nehmen wir folgende Einteilung der Menge $T(Y)$ vor:

Definiton. Die Menge $T(Y)^e$ bestehe aus allen Knoten $s \in T(Y)$, für die kein Knoten $t \in T(Y)$ mit $s \triangleleft^* t$ existiert. Weiter sei $T(Y)^i$ die Menge derjenigen Knoten $s \in T(Y)^e$, für die überhaupt kein Knoten $t \in \mathcal{J}$ mit $s \triangleleft^* t$ existiert. Schließlich sei $T(Y)^a$ die Menge $T(Y)^e \setminus T(Y)^i$.

Jeder Knoten $s \in T(Y)^i$ ist ein Endknoten von \mathcal{J} , für den ein Interpoland $I(t)$ bekannt ist, da er keinen Vorgänger mit dem gleichen Paar besitzt. Auf einen Knoten $t \in T(Y)^a$, der nicht Endknoten von \mathcal{J} ist, wird eine Regel auf $x_2(t)$ angewandt. Ist Y frei, so besteht $T(Y)$ nur aus Endknoten, und es ist $T(Y)^a = \emptyset$ und $T(Y)^e = T(Y)^i$.

Lemma 17. Aus $D(T(Y)^e) \models P$ folgt stets $D(T(Y)) \models P$.

BEWEIS. Die Menge $T(Y)$ ist mit der Relation \triangleleft^* im wesentlichen eine Menge von Bäumen, deren Endknoten gerade die Knoten von $T(Y)^e$ sind. Da für Knoten $s, t_1, \dots, t_n \in T(Y)$, wobei die t_i die Nachfolger von s bezüglich \triangleleft^* sind, aus $K(t_i) \models P$ für alle i stets $K(s) \models P$ folgt, gilt $K(s) \models P$ sogar für alle $s \in T(Y)$. \square

Definition. Ist $T(Y)^i = \{t_1, \dots, t_n\}$ mit $n > 0$, so sei $I(Y)$ die Disjunktion über alle $I(t_i)$, für $n = 0$ sei $I(Y) := 0$.

Lemma 18. Ist $T(Y)^a = \emptyset$, so ist $I(Y)$ Interpoland für das Paar $D(T(Y))/Y$.

BEWEIS. Sei $T(Y)^i = \{t_1, \dots, t_n\}$. Da $D(T(Y)^i)$ genau die $K(t_i)$ als Disjunktionsglieder enthält, und stets $K(t_i) \models I(t_i)$ gilt, gilt auch $D(T(Y)^i) \models I(Y)$. Weil $T(Y)^i = T(Y)^e$ ist, folgt $D(T(Y)) \models I(Y)$ mit Lemma 17. Da ferner $Y \models \neg I(t_i)$ für alle $t_i \in T(Y)^i$ gilt, gilt auch $Y \models \neg I(Y)$. Zudem ist $S(I(t_i)) \subseteq S(x_1(t_i)) \cap S(Y)$ für alle i , somit gilt die Behauptung. \square

Lemma 19. Aus $D(T(Y)^q) \neq P$ folgt $D(T(Y)) \neq [\neg I(Y)?]P$.

BEWEIS. Ist $t \in T(Y)^1$, gilt $K(t) \neq I(Y)$. Ist $t \in T(Y)^q$, gilt $K(t) \neq P$. Ist somit $t \in T(Y)^q = T(Y)^1 \cup T(Y)^q$, so gilt $K(t) \neq I(Y) \vee P$, also auch $K(t) \neq [\neg I(Y)?]P$. Somit gilt $D(T(Y)^q) \neq [\neg I(Y)?]P$, und mit Lemma 17 folgt die Behauptung. \square

Wir definieren nun ein Tableau, bei dem jedem Knoten t ein Paar $D(T(Y))/Y$ oder $D(T(Y)^q)/Y$ sowie eine Zahl $k(t) \in \{1, 2, 3\}$ zugeordnet ist. Wir bezeichnen einen solchen Knoten mit $D(T(Y))/_{k(t)}Y$ oder $D(T(Y)^q)/_{k(t)}Y$. Dabei ist Y stets die zweite Komponente eines Knotens von \mathcal{J}^j .

Definition. Das Tableau \mathcal{J}^k habe die Wurzel $D(T(Y_2))/_1Y_2$. Ein Knoten $D(T(Y))/_1Y$ des Tableaus, für den kein Vorgänger t mit dem gleichen Paar und $k(t)=1$ existiert, hat genau einen Nachfolger, nämlich $D(T(Y))/_2Y$, sonst ist er Endknoten. Ein Knoten $D(T(Y))/_2Y$ hat genau dann einen Nachfolger, nämlich $D(T(Y)^q)/_3Y$, wenn $T(Y)^q$ nichtleer ist; sonst ist er Endknoten. Wir bemerken, daß $T(Y)^q$ leer ist, wenn Y frei ist. Ist $T(Y)^q$ nichtleer, so wird auf jeden Knoten $t \in T(Y)^q$ in \mathcal{J}^j dieselbe Regel auf dieselbe Formel in Y angewandt. Es gibt also Formelmengen Z_1, \dots, Z_n so, daß jeder Knoten $t \in T(Y)^q$ genau n Nachfolger mit den zweiten Komponenten Z_1, \dots, Z_n hat. Dann seien die Nachfolger von $D(T(Y)^q)/_3Y$ gerade die Paare $D(T(Z_1))/_1Z_1, \dots, D(T(Z_n))/_1Z_n$.

Ist K die Anzahl der verschiedenen zweiten Komponenten von \mathcal{J}^j , so ist \mathcal{J}^k nicht länger als $3K+1$.

Zusätzlich definieren wir für alle $s, t \in \mathcal{J}^k$ mit $s \triangleleft t$ ein Programm $\mathcal{P}(s, t)$, das wir das *kanonische Programm* von s zu t nennen. Sind dann s, t beliebige Knoten mit $s \triangleleft t$, ist $s = s_1 \triangleleft \dots \triangleleft s_n = t$ der Pfad von s zu t , und ist $a_i = \mathcal{P}(s_i, s_{i+1})$, $i < n$, das kanonische Programm von s_i zu s_{i+1} , so sei $\mathcal{P}(s, t) := a_1; \dots; a_{n-1}$ das kanonische Programm von s zu t . Für $a = \mathcal{P}(s, t)$ schreiben wir auch $s \xrightarrow{a} t$.

Wenn wir im folgenden vom einen i -Knoten s reden, so meinen wir einen Knoten s von J^k mit $k(s)=i$. Weiter bezeichnen wir mit s^i, t^i, \dots stets beliebige i -Knoten.

Definition. Sei s^j ein Knoten so, daß für alle t, t' mit $s^j \leq t < t'$ das kanonische Programm von t zu t' bereits definiert ist, und sei $s^i < s^j$.

- Ist $i=2$ und $j=3$, so ist s^i ein Paar $D(T(Y))/_2Y$ und s^j ein Paar $D(T(Y)^4)/_3Y$ zugeordnet. Es sei $P(s^i, s^j) := \neg I(Y)?$.
- Ist $i=1$ und $j=2$, so ist s^i ein Paar $D(T(Y))/_1Y$ und s^j ein Paar $D(T(Y))/_2Y$ zugeordnet. Seien t_1, \dots, t_n alle Nachfolger von s^i mit dem gleichen Paar und $k(t_i)=1$ für alle $1 \leq i \leq n$. Ist $n=0$, so sei $P(s^i, s^j) := 1?$. Ist $n > 0$, so sei a_i das kanonische Programm von s^j zu t_i . In diesem Fall sei $P(s^i, s^j) := (a_1 \cup \dots \cup a_n)^*$.
- Ist $i=3$ und $j=1$, so ist s^i ein Paar $D(T(Y)^4)/_3Y$ und s^j ein Paar $D(T(Z))/_1Z$ zugeordnet. Die Menge Z entsteht dabei aus Y durch Anwendung einer gewissen Regel. Handelt es sich dabei nicht um (At) , so sei $P(s^i, s^j) := 1?$. Sonst ist $Y=Y'; \neg(A)P^R$ und $Z=Y'; \neg(A); \neg P^R P^R$ für gewisse Y' und $\neg(A)P^R$. In diesem Fall sei $P(s^i, s^j) := A$.

Ist mit dieser Definition $a=P(s, t)$ und $D(T)/_iY$ das dem Knoten s zugeordnete Paar, so gilt $S(a) \subseteq S(D(T)) \cap S(Y)$.

Lemma 20. Sei $s^i < s^j$. Sei s^i das Paar $D(T)/_iY$ und s^j das Paar $D(T')/_jY'$ zugeordnet. Ist a das kanonische Programm von s^i zu s^j , so folgt aus $D(T') \neq P$ stets $D(T) \neq [a]P$.

BEWEIS. Sei zunächst $s^i < s^j$, und für das bei s^j beginnende Teilttableau von J^k sei die Behauptung bereits bewiesen (Induktionsvoraussetzung).

- Ist $i=2$ und $j=3$, so ist $Y=Y'$, $T=T(Y)$ und $T'=T(Y)^4$, und $a=\neg I(Y)?$. Die Behauptung folgt mit Lemma 19.

- Ist $i=1$ und $j=2$, so ist $Y=Y'$ und $T=T'=T(Y)$. Falls $a=1?$, ist die Behauptung trivial. Sei $a=(a_1 u \dots u a_n)^*$. Es gibt dann Nachfolger t_1, \dots, t_n von s^j , denen das Paar $D(T)/Y$ zugeordnet ist, und für jedes t_k , $k \leq n$, a_k das kanonische Programm von s^j zu t_k ist. Nach Induktionsvoraussetzung folgt $D(T) \models [a_k]P$ für jedes k , also auch $D(T) \models [a_1 u \dots u a_n]P$. Wenn aber $D(T) \models [a_1 u \dots u a_n]P$ aus $D(T) \models P$ folgt, folgt auch $D(T) \models [(a_1 u \dots u a_n)^*]P$.
- Ist $i=3$ und $j=1$, so ist $T=T(Y)^4$ und $T'=T(Y')$. Falls $a=1?$, so ist $K(t)$ für alle $t \in T$ ein Disjunktionsglied von $D(T')$, und es gilt $K(t) \models P$. Damit folgt $D(T) \models P$ und $D(T) \models [1?]P$. Falls $a=A$ für ein atomares Programm A ist, so ist für alle $t \in T$ die Konjunktion über $x_1(t)_A$ ein Disjunktionsglied von $D(T')$, und es gilt $x_1(t) \models [A]P$. Damit folgt $D(T) \models [A]P$.

Ist nun $s^i \Delta s^j$ und $a_1; \dots; a_n = P(s^i, s^j)$, so folgt aus $D(T') \models P$ bereits $D(T) \models [a_1] \dots [a_n]P$, das bedeutet aber auch $D(T) \models [a_1; \dots; a_n]P$. \square

Lemma 21. Für jeden Endknoten von \mathcal{J}^k , für den es keinen Vorgänger mit dem gleichen Paar gibt, gibt es einen Interpolanden. Außerdem hat \mathcal{J}^k mindestens einen derartigen Endknoten.

BEWEIS. Jedem Endknoten von \mathcal{J}^k , für den kein Vorgänger mit dem gleichen Paar existiert, ist ein Paar $D(T(Y))/_2 Y$ zugeordnet für ein Y , für das $T(Y)^4$ leer ist. Nach Lemma 18 ist $I(Y)$ ein Interpoland für dieses Paar.

Das Tableau, welches aus \mathcal{J}^k durch Entfernen jeweils der ersten Komponente jedes Paares und "Herausnehmen" aller 2-Knoten und 3-Knoten entsteht, ist ein Teiltabelleau eines PDL-Tableaus mit der Wurzel Y_2 . Dieses hat mindestens einen Endknoten Y , für den kein Vorgänger mit der gleichen Menge existiert. Damit ist $D(T(Y))/_2 Y$ der gesuchte Endknoten von \mathcal{J}^k . \square

Definition. Sei t Endknoten von \mathcal{J}^* , und sei t das Paar $D(T)/Y$ zugeordnet. Besitzt t keinen Vorgänger mit dem gleichen Paar, sei $I(t) := I(Y)$. Für jeden anderen Endknoten sei $I(t) := 1$.

Sei s nicht Endknoten von \mathcal{J}^* , und seien s_1, \dots, s_n , $n \geq 1$, die direkten Nachfolger von s . Ist $n=1$, $a = \mathcal{P}(s, s_1)$ und $a \neq 1?$, so sei $I(s) := [a]I(s_1)$. Sonst sei $I(s) := I(s_1) \wedge \dots \wedge I(s_n)$. Ist t_0 die Wurzel von \mathcal{J}^* , so sei $I_0 := I(t_0)$.

Wir zeigen nun, daß die so konstruierte Formel I_0 ein Interpoland für Y_1/Y_2 ist.

Lemma 22. Es gilt $S(I_0) \subseteq S(Y_1) \cap S(Y_2)$.

Zunächst gilt $S(D(T)) \subseteq S(Y_1)$ und $S(Y) \subseteq S(Y_2)$ für jedes Paar $D(T)/Y$ in \mathcal{J}^* . Somit gilt für jeden Endknoten t bereits $S(I(t)) \subseteq S(Y_1) \cap S(Y_2)$ und, nach obiger Bemerkung, für jedes kanonische Programm a auch $S(a) \subseteq S(Y_1) \cap S(Y_2)$. Damit gilt auch die Behauptung. \square

Lemma 23. Es gilt $Y_1 \models I$.

Für jeden Endknoten t von \mathcal{J}^* gilt $x_1(t) \models I(t)$ (entweder trivialerweise, wenn $I(t) = 1$ ist, oder nach Lemma 16). Seien s_1, \dots, s_n die direkten Nachfolger von s , und es gelte $x_1(s_i) \models I(s_i)$ für $1 \leq i \leq n$. Ist $n=1$, $a = \mathcal{P}(s, s_1)$ und $I(s) = [a]I(s_1)$, so ist $x_1(s) \models I(s)$ gerade die Aussage von Lemma 20. Ist $n \geq 1$ und $1? = \mathcal{P}(s, s_i)$ für alle i , so folgt aus $x_1(s) \models [1?]I(s_i)$ für alle i $x_1(s) \models I(s_1) \wedge \dots \wedge I(s_n)$, also $x_1(s) \models I(s)$. Deswegen folgt durch Induktion auch $x_1(t_0) \models I_0$. Nun ist aber die Konjunktion über Y_1 ein Disjunktionsglied von $x_1(t_0) = D(T/Y_2)$, somit gilt auch $Y_1 \models I_0$. \square

Definition. Sei $J(t_0) := \emptyset$. Sei für alle $t < s$ bereits $J(t)$ definiert. Dann sei $J(s)$ die kleinste Menge so, daß für alle Knoten t, t' von \mathcal{J}^* mit dem gleichen Paar und $t < s < t'$, und für alle Formeln $P \in J(t) \cup \{I(t)\}$ gilt, daß $(\mathcal{P}(s, t') \mid P) \in J(s)$ ist (für $s = t'$: daß $P \in J(s)$ ist). Ferner sei $K(s) := (I(s)) \cup J(s) \cup x_2(s)$.

Lemma 24. Es gelten:

- a) $K(t_0) = I_0; Y_2$.
- b) Ist t ein Endknoten von J^k , für den kein Vorgänger mit dem gleichen Paar existiert, so ist $K(t)$ inkonsistent.
- c) Sind $t < t'$ Knoten mit dem gleichen Paar, so ist $K(t) \subseteq K(t')$.

BEWEIS. Alle drei Aussagen folgen leicht aus den Definitionen. \square

Lemma 25. Sind s_1, \dots, s_n die direkten Nachfolger des Knotens $s' \in J^k$, so gibt es ein erweitertes Tableau für $K(s')$, das die Endknoten $K(s_1), \dots, K(s_n)$ sowie (eventuell) weitere inkonsistente freie Endknoten besitzt.

BEWEIS. Wir betrachten fünf Fälle. In den ersten drei Fällen ist $i=1$ oder $i=2$, und s' hat genau einen Nachfolger s^{i+1} . Dabei ist stets $x_2(s') = x_2(s^{i+1})$, es ist also dort $J(s^{i+1})$ und $I(s^{i+1})$ aus $K(s')$ zu konstruieren.

Fall 1: Es ist $i=1$, und es gibt m 1-Knoten $t_1, \dots, t_m > s'$ in J^k , denen das gleiche Paar wie s' zugeordnet ist. Dann ist $s' - b^* \rightarrow s^2$, wobei $b = a_1 u \dots u a_m$ und $a_j = P(s^2, t_j)$ für alle $1 \leq j \leq m$ ist. Es ist $I(s') = [b^*]I(s^2)$, und durch Anwendung von $(*)$ entsteht $I(s^2)$ sowie $[b][b^*]I(s^2)$. Sei weiter $P \in J(s^2)$. Es gibt 1-Knoten u, u' in J^k mit dem gleichen Paar, mit $u < s^2 < u'$ und dem kanonischen Programm $s^2 - a \rightarrow u'$ so, daß $P = [a]Q$ für ein $Q \in J(u) \cup I(u)$ ist. Entweder ist $u < s'$, dann ist bereits $[b^*; a]Q \in J(s')$, und P kann aus dieser Formel durch Anwendung von $(*)$ gebildet werden, oder es ist $u = s'$, $u' = t_j$, und $a = a_j$ für ein $1 \leq j \leq m$. Ist $Q \in J(s')$, so ist $Q = [b^*; a_j]R$ für ein gewisses R . Durch Anwenden von $(*)$ erhält man $[b][b^*; a_j]R$, und durch j -maliges Anwenden von (u) erhält man $P = [a_j]Q = [a_j][b^*; a_j]R$. Ist $Q = I(s')$, so erhält man durch j -maliges Anwenden von (u) auf die bereits vorliegende Formel $[b][b^*]I(s^2)$ die Formel $P = [a_j]Q = [a_j][b^*]I(s^2)$.

Fall 2: Es ist $i=1$, und es gibt keinen 1-Knoten $s > s^1$ mit dem gleichen Paar. Dann ist $s^1 - 1? \rightarrow s^2$. Ist $P \in J(s^2)$, so ist $P = [a]Q$ für gewisse a, Q und $[1?; a]Q \in J(s^1)$. Durch Anwendung von $(?)$ erhält man $P = [a]Q$. Weiter entsteht durch Anwendung von $(1?)$ $I(s^2)$ aus $I(s^1) = [1?]I(s^2)$.

Fall 3: Es ist $i=2$, und mit $Y = x_2(s^3)$ ist $s^2 - \neg I(Y)? \rightarrow s^3$. Sei $P \in J(s^3)$. Es ist $P = [a]Q$ für gewisse a, Q , und es ist $[\neg I(Y)?; a]Q \in J(s^2)$. Die Anwendung von $(;)$ liefert $[\neg I(Y)?][a]Q$, und die Anwendung von $(?)$ liefert zwei Nachfolger. Ein Nachfolger enthält $\neg \neg I(Y); Y$ und ist inkonsistent. Auf ihn wird $(M-)$ angewandt, wodurch ein freier inkonsistenter Knoten entsteht. Der andere Nachfolger enthält $P = [a]Q$. Auf gleiche Weise bildet man $I(s^3)$ aus $I(s^2) = [\neg I(Y)?]I(s^3)$.

In den letzten beiden Fällen ist $i=3$. Dann hat s^1 n Nachfolger u_1, \dots, u_n , wobei die $x_2(u_j)$ aus $x_2(s^3)$ durch Anwendung einer Regel entstehen.

Fall 4: Es ist $i=3$, $n=1$, und $x_2(u_1)$ entsteht aus $x_2(s^3)$ durch Anwendung von (At) . Dann ist $s^3 - A? \rightarrow u_1$ für ein atomares Programm. Ist $P \in J(u_1)$, so ist $[A]P \in J(s^3)$, oder es ist $P = [a]Q$ für gewisse a, Q und $[A; a]Q \in J(s^3)$. In diesem Falle erhält man durch Anwendung von $(;)$ die Formel $[A]P = [A][a]Q$. Außerdem ist $I(s^3) = [A]I(u_1)$. Danach erhält man aber durch Anwenden von (At) alle Formeln von $K(u_1)$.

Fall 5: Es ist $i=3$, $n \geq 1$, und man erhält die $x_2(u_j)$ durch Anwendung einer Regel außer (At) auf $x_2(s^3)$. Weiter ist $s^3 - 1? \rightarrow u_j$ für alle $j \leq n$, und ist $P \in J(u_j)$, so ist $[1?]P \in J(s^3)$, oder es ist $P = [a]Q$ für gewisse a, Q und $[1?; a]Q \in J(s^3)$. In jedem Fall erhält man P durch Anwendung von $(1?)$ oder $(?)$. Schließlich ist $I(s^3) = I(u_1) \wedge \dots \wedge I(u_n)$. Durch $(n-1)$ maliges Anwenden von (\wedge) entstehen alle $I(u_j)$.

In jedem Fall erhält man also Knoten K'_1, \dots, K'_n mit $K(s_j) \in K'_j$ für alle $1 \leq j \leq n$. Durch Anwendung von $(F-)$ auf diese Knoten erhält man schließlich die gewünschten Endknoten $K(s_1), \dots, K(s_n)$. \square

Lemma 26. Es gilt $Y_2 \neq \neg I_0$.

BEWEIS. Man konstruiert ein erweitertes Tableau für $I_0; Y_2^-$, in dem man durch Anwendung von (M+) zunächst den Knoten $K(t_0) = I_0; Y_2$ bildet. Ausgehend von diesem Knoten konstruiert man mit dem im Lemma 25 beschriebenen Verfahren schrittweise für alle Knoten $t \in J^*$ die Knoten $K(t)$. Auf diese Weise erhält man schließlich ein Tableau, dessen Endknoten (neben eventuell einigen freien inkonsistenten Knoten) gerade die Knoten $K(t_1), \dots, K(t_n)$ sind, wobei t_1, \dots, t_n die Endknoten von J^* sind. Für jeden Knoten t_i , der keinen Vorgänger mit dem gleichen Paar hat, ist $K(t_i)$ inkonsistent, und ist er nicht selbst schon frei, erhält man durch Anwendung von (M-) einen freien inkonsistenten Knoten. Für jeden Knoten t_i , für den ein Vorgänger t_i' in J^* mit dem gleichen Paar existiert, ist $K(t_i') \subseteq K(t_i)$. Durch Anwendung von (F-) auf $K(t_i)$ erhält man somit einen Knoten, für den ein Vorgänger mit der gleichen Menge existiert. Damit liegt ein erweitertes Tableau für $I_0; Y_2^-$ vor, wobei diese Menge nur inkonsistente erste freie Nachfolger besitzt. Damit ist $I_0; Y_2^-$ inkonsistent, und es gilt $Y_2 \neq I_0$. \square

Lemma 27. Ist $X_1; X_2$ inkonsistent, so gibt es einen Interpolanden für X_1/X_2 .

Zunächst existiert ein geschlossenes Tableau für X_1/X_2 . Für jeden freien normalen Endknoten dieses Tableaus kann man einen Interpolanden angeben. Für jeden Knoten Y_1/Y_2 des Tableaus, für dessen freie Nachfolger Interpolanden bekannt sind, kann man mit den in diesem Abschnitt beschriebenen Verfahren auch einen Interpolanden konstruieren. Also kann man schrittweise für jeden freien Knoten des Tableaus, also auch für X_1/X_2 , einen Interpolanden konstruieren. \square

Mit diesem Lemma ist der Beweis des Interpolationssatzes abgeschlossen.

2.4 Praktische Durchführung einer Konstruktion

Um die Konstruktion eines Interpolanden praktisch durchzuführen, ist es nicht notwendig, alle im vorigen Abschnitt beschriebenen Schritte auszuführen. Zum Beispiel kann bei der Konstruktion des Tableaus J^k völlig auf die explizite Bestimmung der jeweiligen ersten Komponenten der Knoten verzichtet werden. Sie sind einzig eingeführt worden, um die gewünschten Eigenschaften des konstruierten Interpolanden nachweisen zu können.

Um zu demonstrieren, welche Schritte man tatsächlich ausführen muß, konstruieren wir jetzt als Beispiel einen Interpolanden für das Paar X_1/X_2 mit $X_1 = \{[(A;A)^*](p \wedge [A;(B \cup C)] \cup 0)\}$ und $X_2 = \{\neg[A^*](p \vee [C]q)\}$. Wir bemerken, daß $S(X_1) = \{p, A, B, C\}$, $S(X_2) = \{p, q, A, C\}$ und $S(X_1) \cap S(X_2) = \{p, A, C\}$ ist.

Schritt 1: Konstruktion eines geschlossenen Tableaus J für X_1/X_2 .

Auf der folgenden Seite geben wir ein solches Tableau J an. Wir benutzen dort die Abkürzungen $Q := p \wedge [A;(B \cup C)] \cup 0$ und $P := p \vee [C]q$. Ist außerdem eine Komponente eines Knotens identisch mit derselben Komponente seines Vorgängers, deuten wir das durch einen Punkt an und schreiben sie nicht noch einmal hin.

Schritt 2. Konstruktion von Interpolanden für alle Knoten, die keinen belasteten Nachfolger besitzen, der Endknoten ist.

Dies ist nach den Lemmata 15 und 16 ohne weitere Zwischenschritte durchführbar. Wir haben diese Interpolanden neben den jeweiligen Knoten von J vermerkt.

Es befindet sich nun nur noch ein freier Knoten im Tableau, nämlich die Wurzel X_1/X_2 , für den noch kein

	[(A;A)*]Q / ¬[A*]P
	[(A;A)*]Q / ¬[A*]P'
p^([A;(B∪C)]O; [A;A] [(A;A)*]Q / .	.
p^([A;(B∪C)]O; [A][A] [(A;A)*]Q / .	.
p; ([A;(B∪C)]O; [A][A] [(A;A)*]Q / .	.
p; [A][B∪C]O; [A][A] [(A;A)*]Q / .	.
. / ¬(p∨[C]q) I=p	. / ¬[A][A*]P'
. / ¬p∧¬[C]q I=p	
. / ¬p; ¬[C]q I=p	
	[B∪C]O; [A] [(A;A)*]Q / ¬[A*]P'
	[B]O; [C]O; [A] [(A;A)*]Q / ¬[A*]P'
. / ¬(p∨[C]q) I=[C]O	. / ¬[A][A*]P'
. / ¬p∧¬[C]q I=[C]O	
. / ¬p; ¬[C]q I=[C]O	
. / ¬p; ¬[C]q' I=[C]O	
0 / ¬q I=0	
	[(A;A)*]Q / ¬[A*]P'

Ein geschlossenes Tableau für X_1/X_2 .

Interpoland bekannt ist. Die restlichen Schritte haben also das Ziel, einen Interpolanden für den Knoten X_1/X_2 aus denen seiner ersten freien Nachfolger zu bilden.

Schritt 3. Konstruktion von J^i und J^j .

Dazu bedarf es nur der Entscheidung darüber, welche Nachfolger von X_1/X_2 zu J^j gehören. Es sind dies alle normalen Knoten, deren zweite Komponente eine der folgenden Mengen ist:

$$\begin{aligned} Y_1 &:= \{\neg[A^*]P^*\}, \\ Y_2 &:= \{\neg[A][A^*]P^*\}, \\ Y_3 &:= \{\neg P\}. \end{aligned}$$

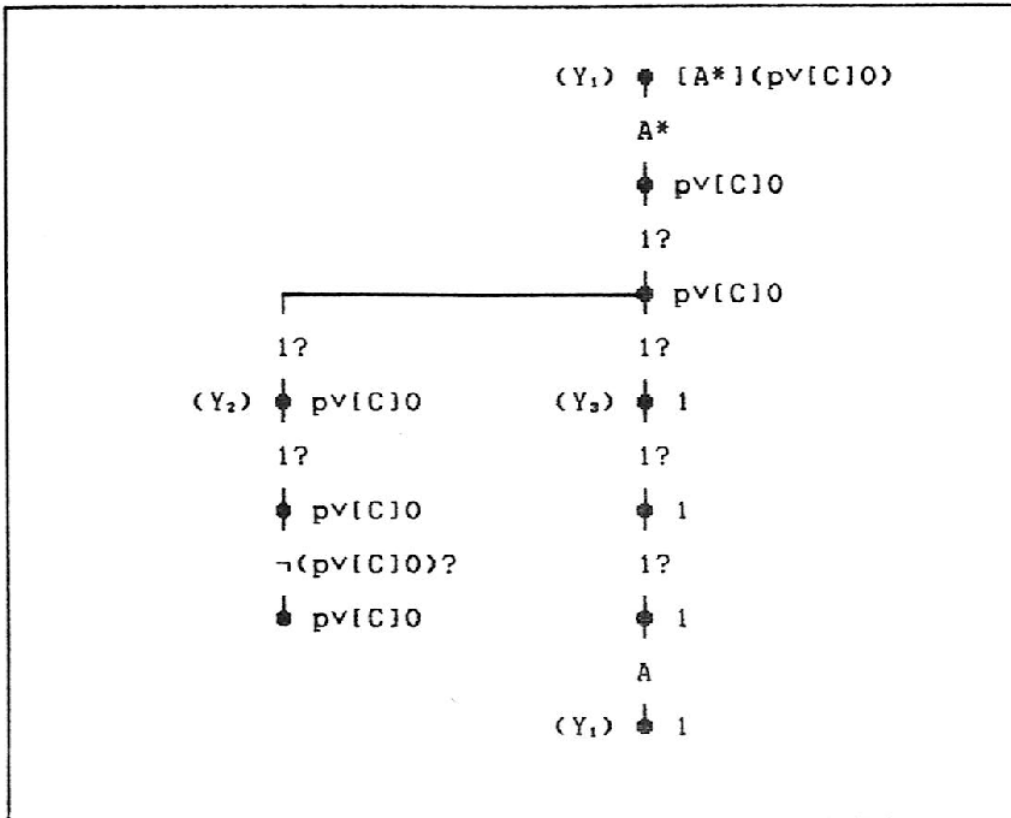
Schritt 4. Bestimmung der notwendigen $I(Y)$.

Alle Knoten von J^j mit der zweiten Komponente Y_3 sind Endknoten von J^j , und für diese sind bereits Interpolanden bekannt (p und $\{C\}$). Es ist somit $T(Y_3)^* = \emptyset$ und $I(Y_3) = p \vee \{C\}$.

Schritt 5. Konstruktion von J^k und I_0 .

Wie schon erwähnt, ist es völlig unnötig, die Formeln $D(T(Y_i))$ usw. explizit zu bestimmen. Die Gestalt des Tableaus J^k hängt ausschließlich von den zweiten Komponenten der Knoten von J^j ab. Zur Konstruktion des Tableaus, das aus den zweiten Komponenten von J^k besteht, reicht es festzustellen, daß Y_1 seine Wurzel ist und die Nachfolger Y_2 und Y_3 hat, Y_1 durch Anwendung von (At) auf $\neg[A][A^*]P^* \in Y_2$ entsteht und Y_3 Endknoten ist.

Man wird feststellen, daß $1?$ ein häufig auftretendes kanonisches Programm in J^k ist. Um einen nicht unnötig großen Interpolanden zu erhalten, wird man günstigerweise bei seiner Konstruktion jedes auftauchende Programm $a;1?$ und $1?;a$ durch das Programm a ersetzen und jede Formel $[\neg P?]P$, $[1?]P$, $1 \wedge P$ oder $P \wedge 1$ durch die Formel P .



Ergebnis der Konstruktion ist also, daß $I_0 = [A^*](p \vee [C]0)$ ein Interpoland für die Formeln $[(A;A)^*](p \wedge [A;(B \cup C)]0)$ und $[A^*](p \vee [C]q)$ ist. Auf Grund der vorgenommenen Überlegungen, einen nicht unnötig großen Interpolanden zu konstruieren, ist I_0 sogar frei von Tests. Dieses Ergebnis ist jedoch nicht immer erreichbar.

Lemma 28. Es gibt Formeln P und Q mit $\models P \rightarrow Q$ so, daß kein Interpoland für P und Q testfrei ist.

BEWEIS. In BERMAN, PATERSON [2] wird nachgewiesen, daß keine testfreie Formel R zur Formel $P := \neg[(p?;A)^*; \neg p?; A; p?]0$ äquivalent ist, d.h. daß es für jede testfreie Formel R ein Kripke-Modell (g, v, n) und ein $S \in g$ mit $S \models P \Leftrightarrow S \models \neg R$ gibt. Damit gilt also nicht sowohl $\models P \rightarrow R$ als auch $\models R \rightarrow P$. Da aber $\models P \rightarrow P$ gilt, heißt das aber, daß keine testfreie Formel Interpoland für P und P ist. \square

2.6 Offene Fragen

Das Ergebnis von Lemma 28 wirft sofort folgende Frage auf:

- Gibt es für alle testfreien Formeln P und Q mit $\models P \rightarrow Q$ einen testfreien Interpolanden ?

Diese Frage würde durch den Nachweis bejaht werden, daß für alle testfreien Formeln P und Q mit $\models P \rightarrow Q$ ein geschlossenes Tableau \mathcal{T} für $P/\neg Q$ so existiert, daß für alle Knoten s^2 und s^3 in \mathcal{T}^* mit $s^2 \triangleleft s^3$ und s^3 nicht Endknoten bereits $\models P(s^2, s^3)$ gilt.

Es erscheint möglich, daß der PDL-Kalkül durch Änderung der kritischen Regel zu einem DPDL-Kalkül und durch Hinzufügung einer weiteren Regel zu einem CPDL-Kalkül gemacht werden kann (beide Logiken sind ebenfalls entscheidbar [5]). Insbesondere interessieren dann folgende Fragen:

- Gilt für DPDL und/oder CPDL ebenfalls der Interpolationssatz, und kann er wie für PDL über einen entsprechenden Tableau-Kalkül bewiesen werden ?

Anhang

Verzeichnis aller Regeln

Die Regeln des PDL-Kalküls:

- | | | |
|---|--|---|
| $(\neg) \frac{X; \neg P}{X; P}$ | $(\wedge) \frac{X; P \wedge Q}{X; P; Q}$ | $(\neg\wedge) \frac{X; \neg(P \wedge Q)}{X; \neg P \mid X; \neg Q}$ |
| $(\neg\cup) \frac{X; \neg(a \cup b)P}{X; \neg[a]P \mid X; \neg[b]P}$ | $(\neg?) \frac{X; \neg[Q?]P}{X; Q; \neg P}$ | $(\neg;) \frac{X; \neg[a; b]P}{X; \neg[a][b]P}$ |
| $(\cup) \frac{X; [a \cup b]P}{X; [a]P; [b]P}$ | $(?) \frac{X; [Q?]P}{X; \neg Q \mid X; P}$ | $(;) \frac{X; [a; b]P}{X; [a][b]P}$ |
| $(\neg n) \frac{X; \neg[a^*]P}{X; \neg P \mid X; \neg[a][a^{(n)}]P}$ | $(n) \frac{X; [a^*]P}{X; P; [a][a^{(n)}]P}$ | |
| $(M+) \frac{X; \neg[a_0] \dots [a_n]P}{X; \neg[a_0] \dots [a_n]P^R} \quad X \text{ frei}$ | <p>die <i>Belastungsregel</i>,</p> | |
| $(M-) \frac{X; \neg[a]P^R}{X; \neg[a]P}$ | <p>die <i>Befreiungsregel</i>,</p> | |
| $(At) \frac{X; \neg[A]P^R}{X_A; \neg P^{R \wedge P}}$ | <p>die <i>kritische Regel</i>.</p> | |
| $(\neg\cup) \frac{X; \neg[a \cup b]P^R}{X; \neg[a]P^R \mid X; \neg[b]P^R}$ | $(\neg;) \frac{X; \neg[a; b]P^R}{X; \neg[a][b]P^R}$ | |
| $(\neg n) \frac{X; \neg[a^*]P^R}{X; \neg P^{R \wedge P} \mid X; \neg[a][a^{(n)}]P^R}$ | $(\neg?) \frac{X; \neg[Q?]P^R}{X; Q; \neg P^{R \wedge P}}$ | |

Die Regeln für die Konstruktion erweiterter PDL-Tableaus:

- | | |
|--|-------------------------------------|
| $(*;) \frac{X; [a^*; b]P}{X; [b]P \mid X; [a][a^*; b]P}$ | $(?;) \frac{X; [! ?; a]P}{X; [a]P}$ |
| $(F-) \frac{X; Y}{X} \quad Y \text{ frei}$ | $(!?) \frac{X; [! ?]P}{X; P}$ |

Literaturverzeichnis

- [1] M.BEN-ARI, J.Y.HALPERN, A.PNUELI, *Deterministic Propositional Dynamic Logic: Finite Models, Complexity, and Completeness*, Journal of Computer and System Sciences 25 (1982), 402-417
- [2] F.BERMAN, M.PATERSON, *Propositional Dynamic Logic is weaker without Tests*, Theoretical Computer Science 16 (1981), 321-328
- [3] K.A.BOWEN, *Interpolation in Loop-free Logic*, Studia Logica 39 (1980), 297-310
- [4] M.J.FISCHER, R.E.LADNER, *Propositional Dynamic Logic of Regular Programs*, Journal of Computer and System Sciences 18 (1979), 194-211
- [5] D.HAREL, *Dynamic Logic*, in "Handbook of Philosophical Logic" Vol.II, 497-604, (D.GABBAY, F.GUENTHNER eds.), Reidel Dordrecht 1984
- [6] G.H.MÜLLER, W.RAUTENBERG (eds.), *Ω -Bibliography of Mathematical Logic*, Vol II, Springer Berlin 1987
- [7] V.R.PRATT, *Semantical Considerations on Floyd-Hoare Logic*, 17th IEEE Symposium on Foundations of Computer Science (1976), 109-121
- [8] V.R.PRATT, *A Practical Decision Method for Propositional Dynamic Logic*, ACM Symposium on Theory of Computing 10 (1978), 326-337
- [9] V.R.PRATT, *Models of Program Logics*, 20th IEEE Symposium on Foundations of Computer Science (1979), 115-122
- [10] V.R.PRATT, *A Near-Optimal Method for Reasoning about Action*, Journal of Computer and System Sciences 20 (1980), 231-254
- [11] W.RAUTENBERG, *Klassische und Nichtklassische Aussagenlogik*, Wiesbaden 1979

- [12] W. RAUTENBERG, *Modal Tableau Calculi and Interpolation*, J.Philosophical Logic 12 (1983), 403-423
- [13] M.Y. VARDI, P. WOLPER, *Automata-Theoretic Techniques for Modal Logic of Programs*, Journal of Computer and System Sciences 32 (1986), 183-221