Kapitel VII:
Modal- und Temporallogik

(Andreas Nonnengart, Hans Jürgen Ohlbach)

1 Modallogiken

Modallogiken sind eine Erweiterung der klassischen Logik um die Konzepte „Zustände“ und „Zustandsübergänge“. Während Prädikatenlogik davon ausgeht, daß – bezüglich einer gegebenen Interpretation – eine Aussage entweder wahr oder falsch ist, erlaubt Modallogik, daß die Interpretation einer Aussage, d.h. ihr Wahrheitswert, sich ändern kann. Darüberhinaus lassen sich über die Veränderung des Wahrheitswertes in Modallogik selbst wieder Aussagen machen und Schlüsse ziehen.

Eine Aussage wie beispielsweise verheiratet(Tom, Mary) ist keine absolute Wahrheit, sondern hängt im allgemeinen von verschiedenen Kontexten ab. Ein solcher ist z.B. der aktuelle Zeitpunkt. Das Prädikat verheiratet(Tom, Mary) kann zwar heute wahr sein, kann aber durchaus wieder falsch werden. Ein anderer denkbarer Kontext ist die Vorstellungs- (oder Traum-)welt eines Beteiligten: verheiratet(Tom, Mary) kann in der Realität falsch, in den Träumen von *Tom* aber wahr sein. Noch ein weiterer möglicher Kontext sind Normen und Rechtslagen: Nach Meinung des Standesamtes kann verheiratet(Tom, Mary) wahr sein, nach Meinung der Kirche aber falsch (weil sie nur standesamtlich getraut sind).

Eine Möglichkeit, diese Kontextabhängigkeit auszudrücken, ist, den Prädikaten mehr Argumente zu geben. Man hätte dann z.B. nicht ein zweistelliges verheiratet-Prädikat, sondern noch zusätzlich Argumente wie Zeit, geistiger Zustand eines Beteiligten (träumend oder nicht), Normlage (Staat, Kirche) etc. Diese Extra-Argumente müssen dabei immer angegeben und in einen geeigneten Kalkül integriert werden.

Modallogiken erlauben dagegen, ohne explizite Angabe eines Kontextes implizit eine Kontextabhängigkeit auszudrücken. Dies geschieht durch die Einführung neuer Operatoren, die zwar syntaktisch dieselbe Rolle wie $\neg$, $\wedge$ usw. spielen, aber eine ganz andere Bedeutung haben. Modallogiken gibt es in einer Vielzahl von Varianten und mit einigen ganz charakteristischen Bedeutungen. Allen gemeinsam ist, daß eine konkrete Interpretation eines Literals den modalen Kontext zu berücksichtigen hat, d.h. die umgebenden Operatoren mit in Betracht ziehen muß.

1.1 Einfache Modallogik

In der einfachsten Modallogik gibt es zusätzlich zu den üblichen logischen Verknüpfungen und Quantoren $\neg$, $\wedge$, $\vee$, $\Rightarrow$ und $\Leftrightarrow$ die beiden einstelligen Operatoren $\Box$ und $\Diamond$ [Che80, HC68]. Traditionell wird der $\Box$-Operator als der *notwendig*-Operator und der $\Diamond$-Operator als der *möglich*-Operator bezeichnet. Formeln werden nicht nur mit den normalen Junktoren und Quantoren gebildet, sondern auch mit diesen neuen Operatoren. Eine zwar nicht allzu sinnvolle, aber syntaktisch erlaubte Formel, ist dann z.B.

$$\Box\ \mathrm{verheiratet(John, Mary)} \lor \forall x\ \Box\ (\exists y\ \Diamond\ \mathrm{verheiratet}(x,y)) \Rightarrow \mathrm{liebt}(x,x).$$

Man kann diese beiden Operatoren also, genau wie die Negation $\neg$, vor beliebige Teilformeln setzen.

Als die Operatoren eingeführt wurden, war ihre intuitive Bedeutung „Φ ist notwendigerweise wahr“ für $\Box\Phi$, sowie „Φ ist möglicherweise wahr“ für $\Diamond\Phi$, um zwischen „zufälligen“ und „nicht zufälligen“ Wahrheiten unterscheiden zu können. Man hoffte, damit viele der damaligen philosophischen Probleme beschreiben, wenn nicht gar lösen zu können. Typische Überlegungen waren dabei von der Art: ist Φ tatsächlich war, wenn Φ notwendigerweise wahr ist, d.h. gilt die Formel $\Box\Phi \Rightarrow \Phi$ oder nicht.[1] Zu Beginn des 20. Jahrhunderts wurden Modaloperatoren dazu verwendet, verschiedene Varianten der strikten Implikation zu beschreiben. Diese strikte Implikation zwischen Φ und Ψ wurde dabei durch die Formel $\Box(\Phi \Rightarrow \Psi)$ definiert und je nach dem, welche Eigenschaften dem $\Box$-Operator zugestanden wurden, erhielt man eine andere Art von strikter Implikation. Diese Eigenschaften wiederum wurden weniger auf semantischer, als auf syntaktischer Ebene definiert und zwar durch das Hinzufügen von Axiomschemata, wie z.B.: $\Box\Phi \Rightarrow \Phi$, $\Box\Phi \Rightarrow \Box\Box\Phi$, $\Box\Phi \Rightarrow \Diamond\Phi$, $\Box(\Phi \Rightarrow \Psi) \Rightarrow (\Box\Phi \Rightarrow \Box\Psi)$, usw.

1959 entwickelte daraufhin Saul Kripke eine modelltheoretische Semantik für die Modallogik, die deren Untersuchung, Anwendung und Automatisierung auf eine neue Ebene rückte [Kri59, Kri63]. Basis dieser Semantik ist Leibniz’ Vorstellung des „Notwendigen“: „Etwas ist notwendigerweise wahr, wenn es in allen vorstellbaren Welten wahr ist.“ Kripke betrachtete also eine Menge von Welten (Zuständen), deren Elementen jeweils eine klassische Interpretation zugeordnet ist. Zusätzlich nahm er eine zweistellige Relation R zwischen Welten an, mit deren Hilfe der Begriff „vorstellbar“ repräsentiert werden soll. Intuitiv bedeutet dann also $R(x,y)$: Die Welt y ist in der Welt x vorstellbar. Heute spricht man von R als der Erreichbarkeitsrelation oder auch Zugriffsrelation. Jeder Welt x entspricht also eine mögliche Interpretation (einem möglichen Zustand) und jede andere, von dieser erreichbaren Welt y, ist in x vorstellbar.

Die Semantik der Modaloperatoren ist jetzt mit Hilfe der Erreichbarkeitsrelation definiert: $\Box\Phi$ ist wahr in einem Zustand z, wenn Φ selbst wahr ist in *allen* von z aus erreichbaren Zuständen. $\Diamond\Phi$ ist wahr in einem Zustand z, wenn Φ selbst wahr ist in *mindestens einem* von z aus erreichbaren Zustand. Die folgenden Beispiele sollen die Intuition dahinter verdeutlichen.

1.1.1 Doxastische Interpretation

In der doxastischen Interpretation verwendet man den $\Box$-Operator, um den „Glauben“ oder die Überzeugungen einer Person (eines Agenten) darstellen zu können. $\Box\Phi$ wird dann also gelesen

[1] Die Konvention ist, daß Φ und Ψ Formelvariablen sind. Sie stehen für beliebige Formeln.

als: Der Agent glaubt Φ.[2]

Nehmen wir beispielsweise die Existenz zweier Vögel an. Einer davon ist der berüchtigte *Tweety* , der andere soll *Hansi* heißen. Nehmen wir desweiteren an, daß der von uns betrachtete Agent glaubt, daß *Tweety* fliegen kann (von *Hansi* weiß er es nicht). Für ihn gibt es also zwei vorstellbare Situationen (Zustände); in der einen fliegen sowohl *Tweety* als auch *Hansi*, in der anderen fliegt zwar *Tweety*, *Hansi* aber nicht. Bildlich kann man dies wie folgt darstellen:

Zustand 1: Fliegt(Tweety) und Fliegt(Hansi)

Realität

Zustand 2: Fliegt(Tweety) und nicht Fliegt(Hansi)

Wir betrachten also insgesamt drei Zustände: die Realität, in der es nicht der Fall sein muß, daß *Tweety*[3] fliegt und die beiden mentalen Zustände 1 und 2 unseres Agenten. Diese werden durch die beiden von der Realität ausgehenden Pfeile veranschaulicht.

In dieser ganz konkreten Interpretation ist also die Formel $\Box \forall x\ \mathsf{Fliegt}(x)$ falsch, da ja *Hansi* im Zustand 2 nicht fliegt; $\Diamond \forall x\ \mathsf{Fliegt}(x)$ ist dagegen wahr, da es einen Zustand gibt, nämlich Zustand 1, in dem beide fliegen. Auch ist $\Box \exists x\ \mathsf{Fliegt}(x)$ wahr, da es ja in beiden Zuständen einen Vogel, nämlich *Tweety*, gibt, der fliegt. Sogar $\exists x\ \Box \mathsf{Fliegt}(x)$ ist in diesem speziellen Fall auch wahr, da wir voraussetzten, daß *Tweety* tatsächlich ein existierender Vogel ist und dieser laut Diagramm in beiden Zuständen fliegt.

Es sei an dieser Stelle noch angemerkt, daß es durchaus auch Zustände geben kann, in denen *Tweety* nicht fliegt. Diese sind aber für unseren Agenten nicht zugänglich.

1.1.2 Epistemische Interpretation

Man kann den $\Box$-Operator auf ganz ähnliche Weise dazu verwenden, auszudrücken, daß man für eine Aussage weiß, ob sie gilt. $\Box\Phi$ bedeutet dann also: Der Agent weiß, daß Φ gilt. Dabei geht man im allgemeinen davon aus, daß man nur Dinge wissen kann, die auch tatsächlich wahr sind und beschreibt dies durch das Formelschema $\Box\Phi \Rightarrow \Phi$, d.h weiß der Agent Φ, so gilt auch Φ in der Realität. Typische Fragestellungen bei dieser Art der Interpretation sind dabei: weiß der Agent, was er weiß (positive Introspektion genannt)? Oder: weiß der Agent, was er nicht weiß (negative Introspektion)? Beides läßt sich mit Hilfe der Modaloperatoren ausdrücken, nämlich durch die beiden Formelschemata $\Box\Phi \Rightarrow \Box\Box\Phi$ und $\neg\Box\Phi \Rightarrow \Box\neg\Box\Phi$.

Die semantische Struktur ist die gleiche wie in der doxastischen Interpretation. Man hat wieder

[2] Wir gehen hier von nur einem Agenten aus; der Ansatz läßt sich aber auch auf mehrere Agenten erweitern.

[3] Wir benutzen generell folgende wichtige Konvention für die Schreibweise: Syntaktische Objekte werden im Zeichensatz wie Tweety geschrieben, während die zugeordneten semantischen Objekte im Zeichensatz wie *Tweety* geschrieben werden. Tweety ist also ein in Termen und Formeln auftauchendes Konstantensymbol, während *Tweety* der reale Tweety ist.

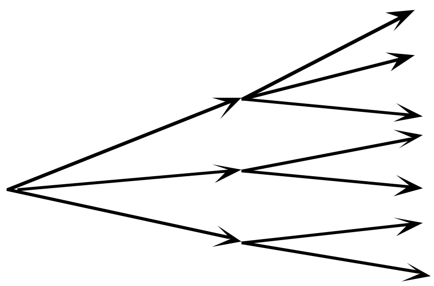
verschiedene Zustände, und $\Box\Phi$ gilt dann, wenn Φ in allen Zuständen, die man in Betracht zieht (alle erreichbaren Zustände), wahr ist. Die Gültigkeit des Axiomenschemas $\Box\Phi \Rightarrow \Phi$ würde dann bedeuten, daß die Realität immer mit in Betracht gezogen wird, d.h. zu den erreichbaren Zuständen gehört.

1.1.3 Temporale Interpretation

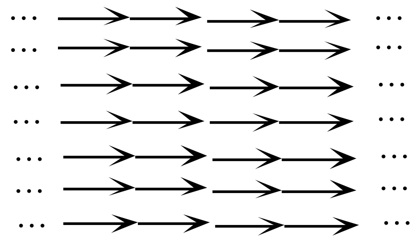
Hier verwendet man $\Box\Phi$ und $\Diamond\Phi$ zum Beispiel in der Bedeutung: Φ gilt immer (in der Zukunft) und: Φ gilt irgendwann (in der Zukunft). Jedem Zeitpunkt entspricht dabei ein Zustand der Welt, in dem bestimmte Fakten wahr sind und andere falsch. Die Erreichbarkeitsrelation ist dann die „früher-später"-Relation, die beliebige Zeitpunkte mit deren zukünftigen Zeitpunkten verbindet.

Typisch für die temporale Interpretation der Modaloperatoren ist dann die Festlegung der Topologie der Zeit. Beispielsweise hat man zu entscheiden, ob man verzweigende oder lineare Zeitstrukturen betrachten möchte. Dabei sollte man nicht den Fehler begehen, lineare Zeit mit Determinismus zu verwechseln. Vielmehr besteht der tatsächliche Unterschied in den verschiedenen natürlich-sprachlichen Interpretationen des $\Diamond$-Operators. In verzweigenden Strukturen ist dieser am besten durch den Ausdruck: „es kann so kommen, daß ..." interpretiert, wohingegen er in linearen Strukturen eher mit dem Begriff „unausweichlich" übersetzt wird. $\Diamond\Phi$ erhält dann also die Bedeutung: „es wird auf jeden Fall so sein, daß Φ", oder anders ausgedrückt: „Φ ist unvermeidlich". In ihrer Mächtigkeit sind lineare und verzweigende Strukturen unvergleichbar, d.h. man kann in verzweigenden Strukturen das „unvermeidlich" und in linearen Strukturen das „möglich" nicht ausdrücken.[4]

Zustände mit verzweigender Zukunft



Zustände mit linearer Zukunft



Abschnitt 1.2. enthält eine etwas tiefergehende Analyse der zeitlichen Interpretation von Modaloperatoren. Wir wollen deshalb an dieser Stelle auf weitere Besonderheiten dieser Interpretation nicht weiter eingehen.

Die hier vorgestellten möglichen Interpretationen bezogen sich alle auf einen $\Box$- und einen $\Diamond$-Operator. In der Praxis ist man allerdings häufig daran interessiert, mehrere verschiedene $\Box$- und $\Diamond$-Operatoren zur Verfügung zu haben. So ist man z.B. in der epistemischen Interpretation

[4]Es gibt Ansätze, durch Hinzunahme weiterer Operatoren diese Lücke für verzweigende Strukturen zu schließen. An dieser Stelle verzichten wir aber auf deren Vorstellung, da diese sich von der klassischen Modallogik etwas entfernen.

nicht nur an der Modellierung eines einzigen Agenten interessiert, sondern vielmehr an einer ganzen Menge von Agenten. Auch ist man in zeitlichen Interpretationen nicht nur an Zukunfts-, sondern auch an Vergangenheitsoperatoren interessiert. Realisieren läßt sich dies durch eine Parametrisierung der Operatoren, so daß nun statt einfach nur $\Box$ und $\Diamond$, beliebige Operatoren der Art $[p]$ und $\langle p \rangle$ zur Verfügung stehen, wobei der Parameter p z.B. Agenten, Zeitrichtungen oder Aktionen bezeichnet. Diese Fälle werden später eingehender behandelt.

1.2 Formale Definition der einfachen Modallogik

1.2.1 Syntax

Die Syntax der Modallogik unterscheidet sich zunächst nicht wesentlich von der, die man von der Prädikatenlogik her kennt. Das heißt, auch hier geht man von einer Signatur Σ, bestehend aus Variablen-, Funktions- und Prädikatensymbolen aus, über die Terme und (unter Hinzunahme von logischen Verknüpfungen und Quantoren) Formeln aufgebaut werden können. Der einzige Unterschied besteht in den beiden zusätzlichen Operatoren $\Box$ und $\Diamond$, für die die folgende Syntaxregel gilt:

Falls Φ eine Formel ist, dann sind $\Box\Phi$ und $\Diamond\Phi$ ebenfalls Formeln.

Als Basis für die Modallogik läßt sich im Prinzip jede beliebige Logik verwenden, wie z.B. Aussagenlogik, Prädikatenlogik erster Stufe (PL1), oder sortierte Prädikatenlogik.

1.2.2 Semantik

Die Semantikdefinition erfolgt in zwei Stufen. Als Grundstock dient die Menge der Zustände und die darauf definierte Erreichbarkeitsrelation. Diese beiden Teile bilden die sogenannten *Frames*. Eine Interpretation erhalten wir dann, wenn wir zusätzlich einen der Zustände als aktuellen Zustand auszeichnen, jeden Zustand mit einer Σ-Struktur (d.i. eine prädikatenlogische Interpretation, welche aus einer Trägermenge besteht und jedem Funktions- und jedem Prädikatensymbol eine passende Funktion bzw. Relation zuordnet) verbinden und zusätzlich eine beliebige Variablenbelegung angeben. Formal heißt dies:

Definition 1 (Semantik der Modallogik)
Ein *Frame* $F = (Z, R)$ besteht aus einer Menge Z von Zuständen und einer binären Relation R auf diesen Zuständen. Eine *Interpretation*[5] $\Im = (F, z, \Theta, V)$ über einer Signatur Σ besteht aus

- einem Frame $F = (Z, R)$,
- einem aktuellen Zustand $z \in Z$,
- einer Funktion Θ, die jedem Zustand eine Σ-Struktur zuordnet und
- einer Variablenbelegung V, die den Variablen Elemente der Trägermenge der Σ-Strukturen zuordnet.

---

[5]Hier ist natürlich die formale logische Interpretation gemeint, nicht die informelle, die wir oben gebraucht haben.

Für einen Zustand z ist dann $\mathfrak{I}_h$ die Abbildung (Σ-Homomorphismus), die jedem Term einen Wert aus der Trägermenge der Σ-Struktur $\Theta(z)$ zuweist (falls einer existiert). Dabei spielen sowohl die Σ-Struktur $\Theta(z)$ wie auch die Variablenbelegung V eine wesentliche Rolle.

Für $\mathfrak{I} = (F, z, \Theta, V)$ sagt man, $\mathfrak{I}$ basiert auf dem Frame F.

Die Erfüllbarkeitsrelation ist dann wie folgt definiert (Erläuterungen siehe unten):

$\mathfrak{I} \models \mathsf{P}(t_1,\ldots,t_n)$	gdw.	$(\mathfrak{I}_h(t_1),\ldots,\mathfrak{I}_h(t_n)) \in \mathsf{P}_{\Theta(z)}$
$\mathfrak{I} \models \neg\Phi$	gdw.	nicht $\mathfrak{I} \models \Phi$
$\mathfrak{I} \models \Phi \wedge \Psi$	gdw.	$\mathfrak{I} \models \Phi$ und $\mathfrak{I} \models \Psi$
$\mathfrak{I} \models \Phi \vee \Psi$	gdw.	$\mathfrak{I} \models \Phi$ oder $\mathfrak{I} \models \Psi$
$\mathfrak{I} \models \forall \mathsf{x}\, \Phi$	gdw.	für alle $x \in$ Trägermenge($\Theta(z)$) gilt $\mathfrak{I}[\mathsf{x}/x] \models \Phi$
$\mathfrak{I} \models \exists \mathsf{x}\, \Phi$	gdw.	es gibt ein $x \in$ Trägermenge($\Theta(z)$) und $\mathfrak{I}[\mathsf{x}/x] \models \Phi$
$\mathfrak{I} \models \Box\Phi$	gdw.	für alle z' mit $R(z,z')$ gilt $\mathfrak{I}[z'] \models \Phi$
$\mathfrak{I} \models \Diamond\Phi$	gdw.	es gibt ein z' mit $R(z,z')$ und $\mathfrak{I}[z'] \models \Phi$.

Dabei verwenden wir die folgenden Konventionen:

Falls $\mathfrak{I} = (F, z, \Theta, V)$ dann ist $\mathfrak{I}[\mathsf{x}/x] := (F, z, \Theta, V[\mathsf{x}/x])$, d.h. $\mathfrak{I}[\mathsf{x}/x]$ unterscheidet sich von $\mathfrak{I}$ nur dadurch, daß die Variable x auf den Wert x abgebildet wird.
Analog ist für einen Zustand z' $\mathfrak{I}[z'] := (F, z', \Theta, V)$, d.h. z' ersetzt den aktuellen Zustand z.

Eine Formel Φ heißt *gültig (erfüllt) in einer Interpretation* $\mathfrak{I}$ falls $\mathfrak{I} \models \Phi$. (Dabei nennen wir den aktuellen Zustand z in $\mathfrak{I}$ den „Startzustand“ für die Interpretation.)

Φ heißt *gültig (erfüllt) in einem Frame F*, geschrieben $F \models \Phi$, falls $\mathfrak{I} \models \Phi$ für *alle F*-basierten Interpretationen $\mathfrak{I}$. ■

Erläuterungen zur Semantikdefinition:

Betrachten wir zunächst den Basisfall: $\mathfrak{I} \models \mathsf{P}(t_1,\ldots,t_n)$ gdw. $(\mathfrak{I}_h(t_1),\ldots,\mathfrak{I}_h(t_n)) \in \mathsf{P}_{\Theta(z)}$.

Um $\mathsf{P}(t_1,\ldots,t_n)$ bezüglich $\mathfrak{I} = (F, z, \Theta, V)$ interpretieren zu können, müssen wir die Werte der Terme $t_1,\ldots,t_n$ in $\mathfrak{I}$ ermitteln und feststellen, welche Relation P in $\mathfrak{I}$ zugeordnet ist. D.h. interpretiert $\mathfrak{I}$ die Terme $t_1,\ldots,t_n$ gerade zu $s_1 = \mathfrak{I}_h(t_1),\ldots,s_n = \mathfrak{I}_h(t_n)$ und ordnet $\mathfrak{I}$ dem Symbol P die Relation $P = \mathsf{P}_{\Theta(z)}$ zu, so gilt $\mathfrak{I} \models \mathsf{P}(t_1,\ldots,t_n)$ genau dann, wenn $(s_1,\ldots,s_n) \in P$. Es ist also notwendig, zunächst beliebige Terme evaluieren zu können. Für Variablen ist dies sehr einfach, da jede Interpretation eine Variablenbelegung mit sich führt. Für komplexe Terme der Form $\mathsf{f}(t_1,\ldots,t_n)$ hingegen, muß als erstes das Funktionssymbol f geeignet interpretiert werden. Dies können wir nur mit Hilfe einer Σ-Struktur durchführen; es muß also die hier zuständige Σ-Struktur gefunden werden. Aus der gegebenen Interpretation können wir ersehen, daß z der aktuelle Zustand ist. Θ ist eine Funktion, welche jeden Zustand in eine Σ-Struktur abbildet. Die aktuelle Σ-Struktur ist somit $\Theta(z)$. Als Σ-Struktur weist $\Theta(z)$ dem Symbol f eine Funktion zu, die wir hier $\mathsf{f}_{\Theta(z)}$ nennen wollen. Diese Funktion $\mathsf{f}_{\Theta(z)}$ ist es gerade nach der wir suchten. Nachdem die Werte $s_1,\ldots,s_n$ berechnet sind (dies geschieht rekursiv über den Aufbau der Terme), kann der Term $\mathsf{f}(t_1,\ldots,t_n)$ evaluiert werden. Auf völlig analoge Weise erhalten wir auch die Relation $\mathsf{P}_{\Theta(z)}$, welche dem Symbol P bezüglich $\mathfrak{I}$ zugeordnet ist und es ergibt sich letztendlich:

$$\Im \models \mathsf{P}(t_1,\ldots,t_n) \text{ gdw. } (\Im_h(t_1),\ldots,\Im_h(t_n)) \in \mathsf{P}_{\Theta(z)}.$$

Die Regeln für die anderen prädikatenlogischen Operatoren unterscheiden sich nicht von denen aus der Prädikatenlogik, nur muß man sich die Trägermenge, die bei der Auswertung der Quantoren betrachtet wird aus dem aktuellen Zustand z, d.h. aus der Σ-Struktur $\Theta(z)$, besorgen. Die Regeln für die Modaloperatoren sind jetzt so zu lesen: $\Im \models \Box\Phi$ gilt genau dann, wenn für alle z' die mittels R von z erreichbar sind, $\Im[z'] \models \Phi$ gilt. $\Box\Phi$ ist also insbesondere dann wahr, wenn es überhaupt keine erreichbaren Zustände gibt, und das für jede Formel Φ, auch für die Formel falsch. (Damit ist $\Box$falsch erfüllbar!) Der $\Diamond$-Operator ist dazu dual: Von dem aktuellen Zustand z ausgehend, prüfe ob es irgendeinen erreichbaren Zustand gibt, in dem Φ wahr ist.

Bei den Funktions- und Prädikatensymbolen unterscheidet man manchmal zwischen *starren* und *flexiblen* Symbolen. Starre Symbole haben in allen Zuständen die gleiche Bedeutung, während flexible Symbole ihre Bedeutung ändern können. Die arithmetischen Funktionen +,* usw. würde man typischerweise als starre Funktionen modellieren, während eine Funktion wie Bundeskanzler-von(…) eine flexible Funktion wäre (z.B. in einer temporalen Interpretation der Logik[6]).

Weiterhin unterscheidet man noch danach, ob die Trägermengen der den Zuständen zugeordneten Σ-Strukturen konstant bleiben *(constant domain)*, mit der Erreichbarkeitsrelation wachsen *(increasing domain)* oder kleiner werden *(decreasing domain)*. Increasing domain bedeutet somit, daß von einem Zustand zu einem erreichbaren keine Objekte verschwinden können (alle sind in gewisser Weise „unsterblich“) während bei decreasing domain nur Objekte verschwinden können, aber keine neu auftauchen dürfen. Macht man keine dieser Annahmen, können Objekte sowohl neu entstehen, wie auch wieder verschwinden (die Trägermengen können dabei auch ganz leer werden). Was in diesem Fall passiert, sieht man an dem Term Vater-von(Tom). In einer temporalen Anwendung der Logik kann dann in einem Zustand (vor seiner Geburt) *Tom* nicht existieren, d.h. der Term Tom kann nicht geeignet interpretiert werden, wohl aber hätte man gerne für den Term Vater-von(Tom) eine Interpretation. Im Zustand nach dem Tod von *Tom*'s Vater hat Tom eine Interpretation, aber für den Term Vater-von(Tom) gibt es keine mehr. Mit der urspünglichen Idee, die Terme als partielle Funktionen der Zustände zu interpretieren, kommt man also nicht weit. Wir werden später sehen, wie dieses Problem gelöst werden kann.

Zunächst zeigen wir jedoch einen einfachen Zusammenhang:

Lemma 2 (Dualität von $\Box$ und $\Diamond$)
Die Äquivalenz $\Diamond\Phi \Leftrightarrow \neg\Box\neg\Phi$ gilt für alle Formeln in allen Frames.
Beweis:
$F \models \Diamond\, \Phi \Leftrightarrow \neg\, \Box \neg\, \Phi$
gdw. für alle F-basierten Interpretationen $\Im$:
$\Im \models (\Diamond\, \Phi \Leftrightarrow \neg\, \Box \neg\, \Phi)$

[6] Der Term Bundeskanzler-von(Deutschland) ändert ja seinen Wert von Zeit zu Zeit.

gdw. $\Im \models \Diamond\, \Phi$ gdw. $\neg\, \Box\neg\Phi$
gdw. (es gibt ein z' mit $R(z,z')$ und $\Im[z'] \models \Phi$)
gdw. nicht (für alle z' mit $R(z,z')$ gilt $\Im[z'] \models \neg\Phi$)
gdw. (es gibt ein z' mit $R(z,z')$ und $\Im[z'] \models \Phi$)
gdw. (es gibt ein z' mit $R(z,z')$ und nicht $\Im[z'] \models \neg\Phi$)
gdw. (es gibt ein z' mit $R(z,z')$ und $\Im[z'] \models \Phi$)
gdw. (es gibt ein z' mit $R(z,z')$ und $\Im[z'] \models \Phi$)
gdw. wahr. ■

Also ist der $\Diamond$-Operator im Prinzip nur eine Abkürzung für $\neg\Box\neg$. $\Box$ und $\Diamond$ sind dual zueinander, so wie man dies von den beiden Quantoren $\forall$ und $\exists$ her schon kennt.

Basierend auf der obigen Semantik läßt sich zeigen, daß das aussagenlogische Fragment dieser einfachen Modallogik durch folgenden Hilbertkalkül[7] charakterisiert wird: Axiome, d.h. von vorneherein als wahr angenommene Formelschemata, bestehen aus allen aussagenlogisch wahren Formeln und dem zusätzlichen Axiom K, mit dem der $\Box$-Operator charakterisiert wird:

$$\text{Axiom K:}\quad \Box(\Phi \Rightarrow \Psi) \Rightarrow (\Box\Phi \Rightarrow \Box\Psi)$$

Schlußregeln sind der Modus Ponens und die sogenannte „Notwendigkeitsregel“:

„Notwendigkeitsregel“: $\dfrac{\Phi}{\Box\Phi}$ Φ ist eine Tautologie

Für die volle Logik kommen noch die entsprechenden Regeln für die Quantoren hinzu. Die „Notwendigkeitsregel“ besagt nichts anders, als daß alle allgemeingültigen Aussagen in allen Welten gelten, drückt also eine gewisse Homogenität aus. Das ist genau was man braucht, wenn man den $\Box$-Operator z.B. temporal interpretiert (immer in der Zukunft gilt …).

Das folgende Beispiel zeigt, daß eine „normale“ Modallogik (d.h. eine Modallogik in der sowohl das Axiom K, als auch die „Notwendigkeitsregel“ und der Modus Ponens gelten), bei der adäquaten natürlich-sprachlichen Interpretation der Modaloperatoren Schwierigkeiten bereiten kann:

Nehmen wir an, wir interpretieren den $\Box$-Operator epistemisch (der Agent weiß, daß …). Dann besagt die „Notwendigkeitsregel“, daß dieser Agent alle gültigen Aussagen weiß. Betrachten wir also die Aussage:

Zermelo_Fränkel_Axiome_der_Mengenlehre $\Rightarrow$ Fermats_letztes_Theorem

und nehmen wir an, daß diese wirklich gültig ist, so besagt die „Notwendigkeitsregel“, daß der von uns betrachtete Agent dies auch weiß. Kennt er die Axiome der Mengenlehre, so können wir mit Hilfe des Axioms K und des Modus Ponens sehr einfach herleiten, daß er weiß, daß Fermats letztes Theorem wahr ist.

Das Problem liegt offensichtlich darin, daß die Logik schon garantiert, daß der Agent den

[7]Ein Hilbertkalkül ist ein Aufzählungsverfahren für alle in einer Logik als wahr zu betrachtenden Formeln. Er startet mit elementaren als wahr zu betrachtenden Formelschemata (Axiome) und wendet Schlußregeln an, um daraus alle weiteren Tautologien zu generieren.

deduktiven Abschluß seines Wissens (Axiom K) und sogar alle Tautologien „Notwendigkeitsregel“ kennt. Im üblichen Sprachgebrauch hat der Begriff des „Wissens“ keine derart starke Bedeutung. Wir verbinden ihn meist mit „wissen und sich dessen bewußt sein“. In dieser Interpretation allerdings würden wir die Gültigkeit des Axioms K und der „Notwendigkeitsregel“ nicht akzeptieren. In einer normalen Modallogik entspricht der $\Box$-Operator also einer sehr starken Form des Wissens, welche implizites Wissen mit einschließt (Wissen über das man sich nicht bewußt ist). Man muß also äußerst vorsichtig sein bei der Auswahl der zur Verfügung stehenden Axiome und Regeln, wenn man bestimmte Interpretationen der Modaloperatoren betrachten möchte.

1.3 Relationale Übersetzung in Prädikatenlogik

Um in Modallogik Schlußfolgerungen zu ziehen und Theoreme zu beweisen, ist der oben angegebene Hilbertkalkül denkbar ungeeignet. Es gibt spezielle Kalküle, in erster Linie Tableauxverfahren [Fit83], die das Problem geschickter angehen. Aber auch diese haben ihre Probleme und sind nur beschränkt anwendbar. Es gibt jedoch einen weiteren Ansatz, der sich als vorteilhafter erwiesen hat. Die Idee die sich dahinter verbirgt besteht darin, modallogische Formeln einfach in Prädikatenlogik zu übersetzen, um daraufhin mit prädikatenlogischen Mitteln weiterzuarbeiten. Damit hat man Modallogik als komfortable und benutzerfreundliche Syntax, für die Operationalisierung aber kann auf bewährte prädikatenlogische Methoden zurückgegriffen werden.

Wie bei Compilern für Programmiersprachen gibt es Übersetzer, die mehr oder weniger effizienten „Code“ erzeugen. Bei Logiken bedeutet dies, daß die übersetzten Formeln groß werden können und somit beim Theorembeweisen meist einen großen Suchraum produzieren.

Das erste Verfahren, das hier vorgestellt werden soll, geht auf Moore [Moo80] oder früher zurück. Es ist eines von der weniger effizienten Art, ist aber dafür sehr einfach und einsichtig. Außerdem läßt es sich zur Erzeugung von Korrespondenzaxiomen benutzen (siehe Abschnitt 1.1.4). Daher wird es schon an dieser Stelle eingeführt.

Der relationale Übersetzer folgt genau der Semantikdefinition (Def. 1) und macht die in den Semantikregeln der Erfüllbarkeitsrelation implizit vorkommenden Bedingungen explizit. Dazu führt man für die Erreichbarkeitsrelation ein zweistelliges Prädikatensymbol R ein. R(x,y) soll dann gerade bedeuten, daß *y* von *x* aus erreichbar ist. Desweiteren gibt man den flexiblen Funktions- und Prädikatensymbolen ein zusätzliches Argument, welches einen „Zustandsterm“ verwaltet. Eine Modalformel wie z.B. $\Box$P, wobei P ein 0-stelliges flexibles Prädikatensymbol ist, wird dann entsprechend der Semantikregel für den $\Box$-Operator mit

$$\Im \models \Box\Phi \text{ gdw. für alle } z' \text{ mit } R(z,z') \text{ gilt } \Im[z'] \models \Phi$$

zu

$$\exists a\ \forall z\ R(a,z) \Rightarrow P'(z)$$

übersetzt. Der Existenzquantor $\exists$a kommt von der Bedingung, daß Modalformeln dann als erfüllbar angenommen werden, wenn sie in *irgendeinem* Zustand gelten. Dieses $\exists$a muß also ganz außen um die übersetzten Formeln geschachtelt werden. Es bezeichnet somit den

Startzustand der Interpretation.

Der $\Diamond$-Operator wird analog übersetzt. Schwierigkeiten können allerdings die Quantoren bereiten, falls die Trägermengen der den Zuständen zugeordenten Σ-Strukturen nicht konstant sind *(varying domain)*. In diesem Falle sind die Mengen, über die die Quantoren quantifizieren, zustandsabhängig, was in PL1 nicht ohne weiteres zu modellieren ist. Das Problem läßt sich umgehen, wenn man als Trägermenge der Zielstruktur, in die übersetzt wird, die Vereinigung aller Trägermengen der einzelnen Zustände nimmt und zusätzlich ein zweistelliges Prädikatensymbol exists einführt, welches angibt, ob der Term in dem jeweiligen Zustand existiert oder nicht. Eine Formel $\forall x\ P(x)$ würde dann übersetzt werden in $\exists a \forall x\ exists(a,x) \Rightarrow P'(a, x)$. Diese Formel drückt dann genau das aus, was vorher implizit in der Semantikdefinition des Allquantors steckte: nur falls x im Startzustand a existiert, soll $P(x)$ gelten. Dieser Trick löst das oben angesprochene Problem mit Vater(Tom) in varying domains.

Existenzquantoren bieten jedoch nicht die einzige Möglicheit, auszudrücken, daß ein Objekt in einem bestimmten Zustand existiert. Dies kann auch implizit geschehen. Zum Beispiel sagt die Aussage reich(König-von(Frankreich)) nicht nur aus, daß der König von Frankreich reich ist, sondern insbesondere, daß der König von Frankreich im aktuellen Zustand überhaupt existiert. Die Übersetzung einer solchen Aussage in PL1 muß daher diese implizite Annahme explizit machen und für jedes Argument eines Atoms ein entsprechendes exists(…) einführen.

Wir definieren jetzt die relationale Übersetzungsfunktion Π_R folgendermaßen.

Definition 3 (Relationale Übersetzung in PL1)
Für eine modallogische Formel Φ sei $\Pi_R(\Phi) := \exists a\ \pi(\Phi,a)$, wobei $\pi(\Phi,z)$ folgendermaßen definiert ist (der Einfachheit halber erlauben wir π auch Terme zu übersetzen):

$\pi(x,z) = x$ falls x eine Variable ist

$\pi(f(t_1,\dots,t_n), z) = f'(z,\pi(t_1,z),\dots,\pi(t_n,z))$ falls f ein Funktionssymbol ist

$\pi(P(t_1,\dots,t_n), z) = exists(z,\pi(t_1,z))\wedge\dots\wedge exists(z, \pi(t_n,z)) \wedge P'(z,\pi(t_1,z),\dots, \pi(t_n,z))$
falls P ein Prädikatensymbol ist

$\pi(\neg\Phi, z) = \neg\pi(\Phi,z)$

$\pi(\Phi \wedge \Psi, z) = \pi(\Phi,z) \wedge \pi(\Psi,z)$

$\pi(\Phi \vee \Psi, z) = \pi(\Phi,z) \vee \pi(\Psi,z)$

$\pi(\forall x\ \Phi, z) = \forall x\ exists(z,x) \Rightarrow \pi(\Phi,z)$

$\pi(\exists x\ \Phi, z) = \exists x\ exists(z,x) \wedge \pi(\Phi,z)$

$\pi(\Box\Phi, z) = \forall z'\ R(z,z') \Rightarrow \pi(\Phi,z')$

$\pi(\Diamond\Phi, z) = \exists z'\ R(z,z') \wedge \pi(\Phi,z')$ ■

Für diese Übersetzungsfunktion Π_R läßt sich zeigen, daß sie korrekt und vollständig ist, d.h. eine modallogische Formel Φ ist genau dann modallogisch erfüllbar, wenn $\Pi_R(\Phi)$ prädikatenlogisch erfüllbar ist [Ohl91].

Beispiel 4 (für relationale Übersetzung)
$\Pi_R(\forall x\ \Box\ P(f(x)))$
$= \exists a\ \pi(\forall x\ \Box P(f(x)), a)$
$= \exists a\ \forall x\ exists(a, x) \Rightarrow \pi(\Box P(f(x)), a)$

$= \exists a\ \forall x\ \text{exists}(a, x) \Rightarrow \forall z\ R(a, z) \Rightarrow \pi(P(f(x)), z)$
$= \exists a\ \forall x\ \text{exists}(a, x) \Rightarrow \forall z\ R(a, z) \Rightarrow (\text{exists}(z, \pi(f(x), z)) \wedge P'(z, \pi(f(x), z)))$
$= \exists a\ \forall x\ \text{exists}(a, x) \Rightarrow \forall z\ R(a, z) \Rightarrow (\text{exists}(z, f'(z, \pi(x,z))) \wedge P'(z, f'(z, \pi(x,z))))$
$= \exists a\ \forall x\ \text{exists}(a, x) \Rightarrow \forall z\ R(a, z) \Rightarrow (\text{exists}(z, f'(z, x)) \wedge P'(z, f'(z, x)))$ ■

In Spezialfällen läßt sich die Übersetzungsfunktion Π_R noch vereinfachen. Z.B. sind in konstanten Domänen die Trägermengen in allen Zuständen gleich. Daher ist das exists-Prädikat immer wahr und kann schon bei der Übersetzung weggelassen werden. Für starre Symbole ist das zusätzliche Zustandsargument überflüssig und kann (für diese Symbole) entfallen.

1.4 Korrespondenzeigenschaften

Die Modallogik, wie sie bisher vorgestellt wurde, bildet nur eine Basisversion (System K). In speziellen Anwendungen erreicht man sehr schnell einen Punkt, an dem man zusätzliche Eigenschaften verlangt. Soll der $\Box$-Operator z.B. temporal interpretiert werden (immer in der Zunkunft … und entsprechend der $\Diamond$-Operator als irgendwann in der Zukunft …), dann hätte man sicherlich gerne die Gültigkeit des folgenden Schemas gewährleistet:

$$\Diamond\Diamond\Phi \Rightarrow \Diamond\Phi$$

D.h., wenn, ausgehend von „jetzt“, von irgendeinem Zeitpunkt in der Zukunft aus ein weiterer Zeitpunkt in der Zukunft existiert in dem Φ gilt, dann gibt es auch von jetzt aus einen Zeitpunkt in der Zukunft, in dem Φ gilt. Was das implizit ausdrückt, ist nichts anderes, als daß die Zukunft der Zukunft selber wieder Zukunft, die früher-später-Relation also transitiv ist. Tatsächlich besteht ein direkter Zusammenhang zwischen dem Axiomschema $\Diamond\Diamond\Phi \Rightarrow \Diamond\Phi$ bzw. dem dazu kontrapositiven Schema[8] $\Box\Phi \Rightarrow \Box\Box\Phi$ und der Transitivität der Erreichbarkeitsrelation (es handelt sich um ein Schema, weil man wieder die Gültigkeit für *alle* Formeln Φ fordert).

Der Zusammenhang zwischen dem Schema $\Box\Phi \Rightarrow \Box\Box\Phi$ und der Transitivität der Erreichbarkeitsrelation läßt sich folgendermaßen darstellen: Erfüllt ein Frame F *alle* Instanzen des Schemas $\Box\Phi \Rightarrow \Box\Box\Phi$, d.h. die Formeln sind wahr in *allen* F-basierten Interpretationen, dann muß die Erreichbarkeitsrelation in F transitiv sein; und umgekehrt, ist die Erreichbarkeitsrelation in F transitiv, so sind alle Instanzen des Schemas in diesem Frame erfüllt.

Dies bedeutet, daß das obige Axiomenschema die Klasse der Frames mit transitiver Erreichbarkeitsrelation *eindeutig* charakterisiert. Diesen Zusammenhang kann man jetzt ausnutzen, indem man das Transitivitätsaxiom für das Prädikat R zu den vom Übersetzer Π_R (Def. 3) erzeugten Formeln hinzufügt und damit wie üblich Beweise führt. Die Theoreme, die sich damit beweisen lassen, sind dann genau diejenigen, die sich unter Annahme des Axiomenschemas $\Box\Phi \Rightarrow \Box\Box\Phi$ beweisen lassen.

[8]Man drehe die Implikation mittels der Kontrapositionsregel $(\Phi \Rightarrow \Psi) \Leftrightarrow (\neg\Psi \Rightarrow \neg\Phi)$ herum, schiebe die Negation unter Ausnutzung von $\neg\Diamond\,\Phi \Leftrightarrow \Box\,\neg\Phi$ nach innen und mache schließlich, da das Φ für eine beliebige Formel steht, aus $\neg\Phi$ einfach ein Φ.

Mittels des Zusammenhangs zwischen dem Axiomenschema und der korrespondierenden Eigenschaft der Erreichbarkeitsrelation sowie der Übersetzung von Modalformeln in Prädikatenlogik läßt sich dann das Rechnen mit dem *Axiomenschema* auf das viel einfachere Rechnen mit dem korrespondierenden *Axiom* zurückführen.

Die Fragen, die sich jetzt stellen, sind:

- Wie entdeckt man solche Korrespondenzen zwischen Axiomenschemata und Eigenschaften der Erreichbarkeitsrelation?
- Wie beweist man die Korrespondenzen?
- Existiert überhaupt für jedes Axiomenschema eine Klasse von Frames mit korrespondierenden Eigenschaften?
- Wenn ja, ist die korrespondierende Eigenschaft in PL1 beschreibbar?
- Existiert für jede Klasse von Frames ein korrespondierendes Axiomenschema?
- Falls es nicht für alle Schemata solche Korrespondenzen gibt, kann man diejenigen syntaktisch charakterisieren, für die es in PL1 beschreibbare Eigenschaften gibt?

Manche Logiker haben sich ausführlich mit diesen Fragen beschäftigt. Ein Überblick dazu gibt [Ben84]. Wir möchten uns an dieser Stelle nicht mit den theoretischen Charakterisierungen beschäftigen, die insbesondere van Benthem in seiner Dissertation gefunden hat, sondern einen Algorithmus angeben, der als Eingabe ein Axiomenschema erhält (und zusätzlich eine Übersetzungsfunktion, z.B. Π_R), und als Ergebnis — falls er terminiert — das zugehörige Korrespondenzaxiom liefert. Die Idee die sich hinter dem SCAN-Algorithmus („Syntesize Correspondence Axioms for Normal logics“) verbirgt ist, das *negierte* Axiomenschema mit dem gegebenen Übersetzer in PL1 zu übersetzen und zu versuchen, es mittels Resolution zu widerlegen. Wenn man das geschickt macht, ergibt sich dabei (per Abduktion) genau die Eigenschaft der Erreichbarkeitsrelation, welche noch fehlt, um eine Widerlegung zu finden. Der Algorithmus funktioniert dabei nicht nur für Modallogik, sondern für alle Logiken mit modelltheoretischer Semantik und einem korrekten und vollständigen Übersetzer in Prädikatenlogik.

Da mit diesem Algorithmus die Bedingungen ausgerechnet werden sollen, unter denen das Axiomenschema beweisbar ist und diese Bedinungen unter Umständen Gleichungen sein können, muß die Resolutionsregel etwas verändert werden. Die Resolution darf nicht scheitern, wenn zwei Terme nicht unifizierbar sind. In diesem Fall muß sie die beiden Terme als Ungleichungen (die mit den gesuchten Gleichungen beweisbar werden) in die Resolvente mit aufnehmen. Die folgende Definition erweitert daher die Resolutions- und Faktorisierungsregel zur C-Resolution und C-Faktorisierung (C für Constraint).

Definition 5 (C-Resolution)

C-Resolution arbeitet folgendermaßen:

$$\frac{\begin{array}{l} P(s_1,\ldots,s_n) \vee C \\ \neg P(t_1,\ldots,t_n) \vee D \end{array}}{C \vee D \vee s_1 \neq t_1 \vee \ldots \vee s_n \neq t_n}$$

$P(s_1,\ldots,s_n)$ und $\neg P(t_1,\ldots,t_n)$ sind die *Resolutionsliterale*

(oder $s_1 = t_1 \wedge \ldots \wedge s_n = t_n \Rightarrow (C \vee D)$)

und die C-Faktorisierung ganz analog:

$$\underline{P(s_1,\ldots,s_n) \vee P(t_1,\ldots,t_n) \vee C}$$

$P(s_1,\ldots,s_n) \vee C \vee s_1 \neq t_1 \vee \ldots \vee \ s_n \neq t_n$ ■

Definition 6 (Algorithmus SCAN)

Eingabe: Ein Axiomschema **A** mit Formelvariablen **P** sowie eine korrekte und vollständige Übersetzungsfunktion in Prädikatenlogik.

Ausgabe: (im Erfolgsfall) Ein Korrependonzaxiom, d.h. eine PL1-Formel, die die Formelvariablen **P** nicht mehr enthält und die genau die Bedingung darstellt, die man braucht, um **A** zu beweisen.

1. Schritt: Negiere **A**.
2. Schritt: Übersetze $\neg$**A** in Prädikatenlogik. (Die n-stelligen Formelvariablen Φ werden dabei als n-stellige Prädikatensymbole P betrachtet).
   Erzeuge die Klauselnormalform.
3. Schritt: C-resolviere und C-faktorisiere auf den Literalen der übersetzten Formelvariablen, solange es geht. Sind alle Resolutionsmöglichkeiten mit einem Literal erschöpft, lösche die Klausel (Puritylöschung). Mache alle in Resolutionsbeweisern üblichen Vereinfachungen auf der Klauselmenge (Tautologie-, Subsumptionslöschung usw).
4. Schritt: Falls der 3. Schritt terminiert und die Klauselmenge nicht leer ist: Mache, falls möglich, die Skolemisierung der Existenzquantoren wieder rückgängig.
5. Schritt: Negiere das Resultat. ■

Der Algorithmus kann an drei Stellen scheitern: Erstens kann die Klauselmenge durch Puritylöschung zur leeren Klauselmenge kollabieren. In diesem Fall existiert definitiv keine Klasse von korrespondierenden Frames, die ohne einen Quantor $\forall\Phi$ in Logik 2. Ordnung beschrieben werden kann. Zweitens, das Verfahren muß nicht in jedem Falle terminieren. Das kann zwei Ursachen haben: entweder ist die Resolution in redundanten Schleifen gefangen und produziert nichts Neues, oder sie produziert unendlich viele neue, aber verschiedene Klauseln. Im letzten Fall gibt es zwar eine Klasse von korrespondierenden Frames, aber die ist nicht mit PL1-Axiomen beschreibbar. An Lemma 43 sieht man, daß man aber u.U. dann Formeln in Logik 2. Ordnung berechnen kann. Leider kann man den Fall, daß sich die Resolution in redundanten Schleifen verfangen hat, nicht automatisch erkennen. Daher ist der Algorithmus insgesamt unvollständig. Wenn man den Algorithmus von Hand durchrechnet, hat man aber gerade damit im allgemeinen wenig Probleme.

Die dritte und letzte Stelle, an der der Algorithmus scheitern kann, ist der Versuch, die Skolemisierung rückgängig zu machen (Entskolemisierung). Falls dies nicht geht, gibt es zwar eine Klasse von korrespondierenden Frames, diese ist aber, wie im zweiten Falle, nicht in PL1 beschreibbar. Benutzt man jedoch Prädikatenlogik 2. Ordnung mit Funktionsvariablen, sowie Henkin-Quantoren[9] [Hen61, Wes89], lassen sich die Formeln in jedem Fall entskolemisieren und man erhält Formeln 2. Ordnung als Korrespondenzaxiome.

In [GO92] wird gezeigt, daß der Algorithmus korrekt ist, d.h. liefert er eine Formel, so stellt diese gerade das gesuchte Korrespondenzaxiom dar.

---

[9]Das sind parallele Kombinationen von $\forall\exists$ Quantoren.

Wir wenden den Algorithmus jetzt auf verschiedene Axiomenschemata an und zeigen damit die in der Literatur vielfach zitierten Korrespondenzen. Dabei müssen im 3. Schritt des Algorithmus immer alle Literale bis auf die R-Literale wegresolviert werden. Mit den R-Literalen wird nur in den wenigen Fällen resolviert, in denen man die Klauseln kürzer machen kann, ohne sie zu instantiieren. (Da wir mit syntaktischen Mitteln Semantik berechnen, benutzen wir diesmal einen neutralen Zeichensatz.)

Beispiel 7	**(Reflexivität)**
Axiomenschema	$\Box\Phi \Rightarrow \Phi$
negiert:	$\Box\Phi \wedge \neg\Phi$
übersetzt:	$\exists a\ (\forall x\ R(a,x) \Rightarrow P(x)) \wedge \neg P(a)$
Klauselform:	$\neg R(a,x),\ P(x)$
	$\neg P(a)$
P wegresolviert:	$\neg R(a,a)$
entskolemisiert:	$\exists a\ \neg R(a,a)$
negiert:	$\forall x\ R(x,x)$ (Reflexivität) ■

Beispiel 8	**(Transitivität)**
Axiomenschema	$\Box\Phi \Rightarrow \Box\Box\Phi$
negiert:	$\Box\Phi \wedge \Diamond\Diamond\neg\Phi$
übersetzt:	$\exists a\ (\forall x\ R(a,x) \Rightarrow P(x)) \wedge \exists b\ R(a,b) \wedge \exists c\ R(b,c) \wedge \neg P(c)$
Klauselform:	$\neg R(a,x),\ P(x)$
	$R(a,b)$
	$R(b,c)$
	$\neg P(c)$
P wegresolviert:	$\neg R(a,c)$
	$R(a,b)$
	$R(b,c)$
entskolemisiert:	$\exists a,b,c\ \neg R(a,c) \wedge R(a,b) \wedge R(b,c)$
negiert:	$\forall x,y,z\ R(x,y) \wedge R(y,z) \Rightarrow R(x,z)$ (Transitivität) ■

Beispiel 9	**(Konfluenz)**
Axiomenschema	$\Diamond\Box\Phi \Rightarrow \Box\Diamond\Phi$
negiert:	$\Diamond\Box\Phi \wedge \Diamond\Box\neg\Phi$
übersetzt:	$\exists a\ (\exists b\ R(a,b) \wedge (\forall x\ R(b,x) \Rightarrow P(x)) \wedge$
	$(\exists c\ R(a,c) \wedge (\forall y\ R(c,y) \Rightarrow \neg P(y))$
Klauselform:	$R(a,b)$
	$\neg R(b,x),\ P(x)$
	$R(a,c)$

$\neg R(c,y),\ \neg P(y)$

P wegresolviert: $R(a,b)$

$R(a,c)$

$\neg R(b,x),\ \neg R(c,x)$

entskolemisiert: $\exists a,b,c\ R(a,b) \wedge R(a,c) \wedge (\forall x\ \neg R(b,x) \vee \neg R(c,x))$

negiert: $\forall x,y,z\ R(x,y) \wedge R(x,z) \Rightarrow (\exists v\ R(y,v) \wedge R(z,v))$ (Konfluenz) ■

Beispiel 10 **(Distribution über $\vee$)**

Axiomenschema $\Box(\Phi \vee \Psi) \Rightarrow (\Box\Phi \vee \Box\Psi)$ (ist äquivalent zu $\Diamond\Phi \Rightarrow \Box\Phi$)

negiert: $\Box(\Phi \vee \Psi) \wedge (\Diamond\neg\Phi \wedge \Diamond\neg\Psi)$

übersetzt: $\exists a\ \forall x\ R(a,x) \Rightarrow (P(x) \vee Q(x)) \wedge$

$\exists b\ R(a,b) \wedge \neg P(b) \wedge$

$\exists c\ R(a,c) \wedge \neg Q(c)$

Klauselform: $\neg R(a,x),\ P(x),\ Q(x)$

$R(a,b)$

$\neg P(b)$

$R(a,c)$

$\neg Q(c)$

P, Q wegresolviert: $R(a,b)$

$R(a,c)$

$b \neq c$ (Hier ist einmal mit R(a,b) resolviert worden.)

entskolemisiert: $\exists a,b,c\ R(a,b) \wedge R(a,c) \wedge b \neq c$

negiert: $\forall x,y,z\ (R(x,y) \wedge R(x,z)) \Rightarrow y = z$ (R ist partielle Funktion) ■

Beispiel 11 **(Geachs Axiom, Rechtslinearität)**

Axiomenschema $\Box(\Box\Phi \Rightarrow \Psi) \vee \Box(\Box\Psi \Rightarrow \Phi)$

negiert: $\Diamond(\Box\Phi \wedge \neg\Psi) \wedge \Diamond(\Box\Psi \wedge \neg\Phi)$

übersetzt: $\exists a\ \exists b\ R(a,b) \wedge (\forall x\ R(b,x) \Rightarrow P(x)) \wedge \neg Q(b) \wedge$

$\exists c\ R(a,c) \wedge (\forall y\ R(c,y) \Rightarrow Q(y)) \wedge \neg P(c)$

Klauselform: $R(a,b)$

$\neg R(b,x),\ P(x)$

$\neg Q(b)$

$R(a,c)$

$\neg R(c,y),\ Q(y)$

$\neg P(c)$

P, Q wegresolviert: $R(a,b)$

$\neg R(b,c)$

$R(a,c)$

$\neg R(c,b)$

entskolemisiert: $\exists a,b,c\ R(a,b) \wedge R(a,c) \wedge \neg R(b,c) \wedge \neg R(c,b)$

negiert: $\forall x,y,z\ R(x,y) \wedge R(x,z) \Rightarrow (R(y,z) \vee R(z,y))$ ■

Beispiel 12 **(Isolierung)**

Axiomenschema $\Phi \Rightarrow \Box\Phi$

negiert: $\Phi \wedge \Diamond\neg\Phi$

übersetzt: $\exists a\ P(a) \wedge \exists b\ R(a,b) \wedge \neg P(b)$

Klauselform: $P(a)$

$R(a,b)$

$\neg P(b)$

P wegresolviert: $R(a,b)$

$a \neq b$

entskolemisiert: $\exists a,b\ R(a,b) \wedge a \neq b$

negiert: $\forall x,y\ R(x,y) \Rightarrow x = y$ ■

Beispiel 13 **(Sackgassen)**

Axiomenschema $\Box$falsch

negiert: $\Diamond$wahr

übersetzt: $\exists a,b\ R(a,b) \wedge$ wahr

Klauselform: $R(a,b)$

entskolemisiert: $\exists a,b\ R(a,b)$

negiert: $\forall x\ \neg \exists y\ R(x,y)$ ■

Beispiel 14 **(Brouwer Axiom)**

Axiomenschema $\Phi \Rightarrow \Box\Diamond\Phi$

negiert: $\Phi \wedge \Diamond\Box\neg\Phi$

übersetzt: $\exists a\ P(a) \wedge \exists b\ R(a,b) \wedge (\forall x\ R(b,x) \Rightarrow \neg P(x))$

Klauselform: $P(a)$

$R(a,b)$

$\neg R(b,x),\ \neg P(x)$

P wegresolviert: $R(a,b)$

$\neg R(b,a)$

entskolemisiert: $\exists a,b\ R(a,b) \wedge \neg R(b,a)$

negiert: $\forall x,y\ R(x,y) \Rightarrow R(y,x)$ (Symmetrie) ■

Beispiel 15 **(Inverse Barcan Formel)**

Axiomenschema $\Box\forall x\ \Phi(x) \Rightarrow \forall x\ \Box\Phi(x)$

negiert: $\Box\forall x\ \Phi(x) \wedge \exists x \Diamond\ \neg\Phi(x)$

übersetzt: $\exists a\ \forall y\ R(a,y) \Rightarrow (\forall x\ \mathrm{exists}(y,x) \Rightarrow P(y,x)) \wedge$

$$\exists x \; exists(a,x) \wedge \; \exists b \; R(a,b) \wedge \neg P(b,x)$$

Klauselform:	$\neg R(a,y), \; \neg exists(y,x), \; P(y,x)$
	$exists(a,c)$
	$R(a,b)$
	$\neg P(b,c)$
P wegresolviert:	$R(a,b)$
	$exists(a,c)$
	$\neg exists(b,c)$ (Hier ist wieder mit R(a,b) resolviert worden.)
entskolemisiert:	$\exists a,b,c \; R(a,b) \wedge \neg \; exists(b,c) \wedge exists(a,c)$
negiert:	$\forall x,y \; R(x,y) \Rightarrow (\forall z \; exists(x,z) \Rightarrow exists(y,z))$ (Increasing Domain) ■

Betrachtet man die Umkehrung der Implikation im obigen Beispiel, d.h. die Barcan-Formel $\forall x \; \Box\Phi(x) \Rightarrow \Box\forall x \; \Phi(x)$, so ergibt sich die „decreasing domain“-Bedingung. Gelten beide Barcan Formeln, dann hat man die Version der Logik mit constant domain, d.h. die Trägermengen sind in allen Zuständen die gleichen.

Gegenbeispiele

Die folgenden Beispiele zeigen, was passieren kann, wenn das Axiomenschema keine in PL1 charakterisierbaren Frames beschreibt.

Beispiel 16	**(Kollaps)**
Axiomenschema	$\Box\Phi \Rightarrow \neg \; \Phi$ (Lügner Axiom)
negiert:	$\Box\Phi \wedge \Phi$
übersetzt:	$\exists a \; (\forall x \; R(a,x) \Rightarrow P(x)) \wedge P(a)$
Klauselform:	$\neg R(a,x), \; P(x)$
	$P(a)$
P wegresolviert:	leere Klausel*menge* ■

Der Kollaps der Klauseln zur leeren Klauselmenge bedeutet, daß es überhaupt keine Bedingung an die Erreichbarkeitsrelation gibt, die $\Box\Phi \Rightarrow \neg\Phi$ beweisbar macht. Um die beiden positiven P-Literale loszuwerden, braucht man negative P-Literale. Die einzige Chance doch noch mit diesem Schema umgehen zu können, liegt darin, die Semantik des $\Box$-Operators zu ändern. Mit der folgenden Version erhalten wir tatsächlich ein brauchbares Resultat:

$$\Im[z] \models \Box\Phi \text{ gdw. für jeden Zustand } z'\text{: } R(z,z') \text{ genau dann wenn } \Im[z'] \models \Phi$$

Der SCAN-Algorithmus muß jetzt mit einer der geänderten Semantik entsprechenden Übersetzungsfunktion angewandt werden.

Beispiel 17	**(Irreflexivität)**
Axiomenschema	$\Box\Phi \Rightarrow \neg\Phi$
negiert:	$\Box\Phi \wedge \Phi$

übersetzt:	$\exists a\ (\forall x\ R(a,x) \Leftrightarrow P(x)) \wedge P(a)$
Klauselform:	$\neg R(a,x),\ P(x)$
	$R(a,x),\ \neg P(x)$
	$P(a)$
P wegresolviert:	$R(a,a)$
entskolemisiert:	$\exists a\ R(a,a)$
negiert:	$\forall x\ \neg R(x,x)$ (Irreflexivität) ■

Beim nächsten Beispiel wird wieder die ursprünglische Semantik verwendet.

Beispiel 18 **(Löbs Axiom)**

Axiomenschema	$\Diamond\Phi \Rightarrow \Diamond(\Phi \wedge \Box\neg\Phi)$
negiert:	$\Diamond\Phi \wedge \Box(\neg\Phi \vee \Diamond\Phi)$
übersetzt:	$\exists a\ \exists b\ R(a,b) \wedge P(b) \wedge$
	$\forall x\ R(a,x) \Rightarrow (\neg P(x) \vee \exists y\ (R(x,y) \wedge P(y)))$
Klauselform:	$R(a,b)$
	$P(b)$
	$\neg R(a,x),\ \neg P(x),\ R(x,f(x))$
	$\neg R(a,x),\ \neg P(x),\ P(f(x))$ ■

Die letzte Klausel ist rekursiv. Daher terminiert die Resolution nicht. Ein Versuch, die P-Literale wegzuresolvieren, produziert unendlich viele Klauseln $R(f^n(b), f^{n+1}(b))$ wobei $f^n(b)$ eine n-fache Schachtelung von f bedeutet. Frames, die diese Klauseln widerlegen, sind Frames mit endlichen *R*-Ketten, und genau das ist Bedingung an die Eigenschaften von *R*, die Löbs Axiom entsprechen (Modallogik G). Diese Bedingung ist jedoch nicht in PL1 axiomatisierbar. Mit der SCAN-Methode kann man sie aber offensichtlich trotzdem (in einem unendlichen Prozeß) ausrechnen. (Dies ist aber noch Gegenstand der Untersuchung.)

Beispiel 19 **(McKinseys Axiom)**

Axiomenschema	$\Box\Diamond\Phi \Rightarrow \Diamond\Box\Phi$
negiert:	$\Box\Diamond\Phi \wedge \Box\Diamond\neg\Phi$
übersetzt:	$\exists a\ \forall x\ R(a,x) \Rightarrow \exists y\ (R(x,y) \wedge P(y)) \wedge$
	$\forall x\ R(a,x) \Rightarrow \exists y\ (R(x,y) \wedge \neg P(y))$
Klauselform:	$\neg R(a,x),\ R(x,f(x))$
	$\neg R(a,x),\ P(f(x))$
	$\neg R(a,y),\ R(y,g(y))$
	$\neg R(a,y),\ \neg P(g(y))$
P wegresolviert:	$\neg R(a,x),\ R(x,f(x))$
	$\neg R(a,y),\ R(y,g(y))$
	$\neg R(a,x),\ \neg R(a,y),\ f(x) \neq f(y)$ ■

Die letzte Klausel kann nur mit einem Henkin-Quantor, einer Art parallele Kombination von $\forall$

$\exists$ Sequenzen, entskolemisiert werden [Hen61]. Logiken mit Henkin-Quantoren sind jedoch echt stärker als PL1, und in der Tat charakterisiert McKinseys Axiom keine in PL1 axiomatisierbaren Frames [Ben84, Seite 197].

Die Klasse der Frames mit gleicher Eigenschaft, z.B. Reflexivität oder Transitivität der Erreichbarkeitsrelation, faßt man üblicherweise zu einer speziellen Logik zusammen, indem man den Erfüllbarkeitsbegriff auf die jeweilige Klasse bezieht. Man sagt dann z.B. eine Formel ist „erfüllbar in der Logik T“ wenn sie in allen Frames mit reflexiver Erreichbarkeitsrelation erfüllbar ist. Die speziellen Logiken, die einer Klasse von Frames mit gleicher Eigenschaft der Erreichbarkeitsrelation entsprechen, haben traditionell folgende Namen:

Definition 20 (Benennung klassischer Modallogiken)

K	keine speziellen Eigenschaften
D	Serialität, d.h. von jedem Zustand gibt es weitere erreichbare Zustände.
T	Reflexivität (manchmal auch M genannt)
DB	Symmtrie und Serialität
K4	Transitivität
B	Reflexivität und Symmetrie
S4	Reflexivität und Transitivität
S5	Reflexivität und Transitivität und Symmetrie (d.h.Äquivalenzrelation)

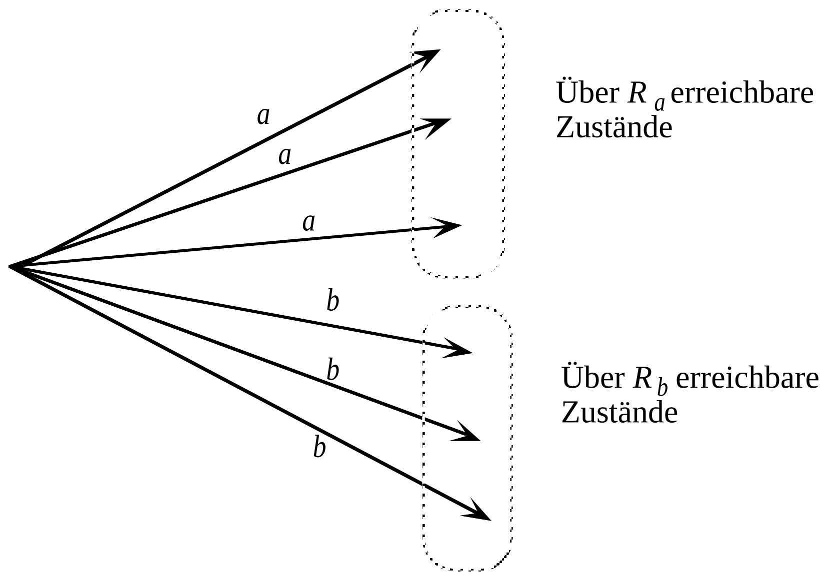
usw. ■

1.5 Modallogik mit parametrisierten Modaloperatoren

Für viele Anwendungen sind die einfachen Modaloperatoren nicht ausdrucksstark genug. Um z.B. eine etwas realistischere epistemische Logik zu bekommen, in der man mehrere „Agenten“ betrachtet, benötigt man parametrisierte Operatoren. $[\mathsf{a}]\mathsf{P}$ könnte dann heißen: Agent a weiß P. In einer Anwendung als Aktionslogik könnte dann $[\mathsf{a}]\mathsf{P}$ heißen: Nach Durchführung der Aktion a gilt P. Diese Parameter der Modaloperatoren können also beliebige Terme sein, die z.B. Agenten oder Aktionen bezeichnen.

Die Semantik ist dabei wieder eine Mögliche-Welten-Semantik mit einer geeigneten Erreichbarkeitsrelation. Allerdings ist diese Erreichbarkeitsrelation jetzt parametrisiert, d.h. für jeden Parameter existiert eine eigene Erreichbarkeitsrelation. Eine Formel $[\mathsf{p}]\mathsf{P}$ wird dann konkret so ausgewertet: Wähle die Erreichbarkeitsrelation R_p und teste ob in allen über R_p erreichbaren Zuständen P gilt. (Für den dualen Operator $\langle\mathsf{p}\rangle$ gilt das Analoge.)

Graphisch kann man sich das so veranschaulichen: Angenommen wir haben zwei Parameter a und b, die von der Interpretation $\mathfrak{I}$ auf die Werte a und b der Trägermenge des aktuellen Zustands abgebildet werden. Dann könnte eine Zustandsstruktur so aussehen:



Die formale Semantik der Modallogik mit parametrisierten Operatoren ist jetzt fast identisch mit der für die einfache Relation. Frames bestehen jetzt aus Zuständen sowie einer parametrisierten Erreichbarkeitsrelation. Damit ist sie eigentlich eine dreistellige Relation.

Definition 21 (Semantik der Modallogik mit parametrisierten Operatoren)
Ein Frame $F = (Z, R)$ besteht aus einer Menge Z von Zuständen und einer binären Relation auf den Zuständen, *die mit Elementen der Trägermenge parametrisiert ist*. Interpretationen sind genauso definiert wie in Def. 1.

Die Erfüllbarkeitsrelation für eine Interpretation mit aktuellen Zustand z sieht jetzt folgendermaßen aus:

$\mathfrak{I} \models [\mathsf{p}]\Phi$	gdw.	für alle z' mit $R_{\mathfrak{I}_h(\mathsf{p})}(z,z')$ gilt	$\mathfrak{I}[z'] \models \Phi$
$\mathfrak{I} \models \langle\mathsf{p}\rangle\Phi$	gdw.	es gibt ein z' mit $R_{\mathfrak{I}_h(\mathsf{p})}(z,z')$ und	$\mathfrak{I}[z'] \models \Phi$

Alle anderen Fälle sind im wesentlichen identisch zum unparametrisierten Fall (Def. 1). Hierbei bezeichnet in $R_{\mathfrak{I}_h(\mathsf{p})}(z,z')$ der Index $\mathfrak{I}_h(\mathsf{p})$ gerade das Element p der Trägermenge des aktuellen Zustands z, zu dem der Parameter p ausgewertet wird. ■

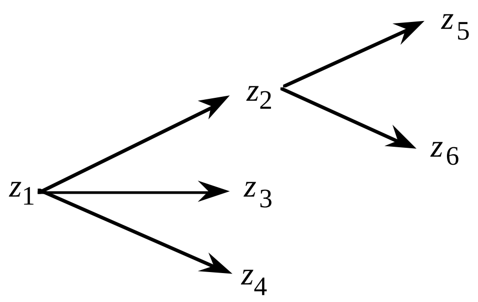
1.6 Funktionale Übersetzung in Prädikatenlogik

Die relationale Übersetzung bewirkt durch die zusätzlichen R-Literale einen erheblichen Ballast innerhalb der Formeln. Diese werden größer und die Information über die Struktur des Zustandsgraphen wird unkontrollierbar verteilt über die gesamte Klauselmenge. Wir stellen daher eine weitere Übersetzungsmethode vor, die erheblich kompaktere Formeln generiert [Her89,Ohl91]. Sie ist deshalb für die praktische Anwendung weit besser geeignet. Außerdem kann sie auch ohne weiteres verwendet werden, um mit Hilfe des SCAN-Algorithmus (Def. 6) die Logiken mit parametrisierten Modaloperatoren zu untersuchen.

In der *funktionalen* Übersetzung erscheinen in den übersetzten Formeln nicht mehr Literale mit dem R-Prädikat, sondern Sequenzen von Termen, welche Übergänge von einem Zustand zu einem nächsten darstellen. Bezeichnet man bei gegebener zweistelliger Relation R für die Menge all der y, die von x aus über R erreichbar sind den Ausdruck $R(x)$, so ergibt sich folgender Zusammenhang:

$$\begin{aligned} R(x) &= \{y \mid R(x,y)\} \qquad (= \{y \mid y \in R(x)\}) \\ &= \{\gamma'(R(x)) \mid \gamma' \in \mathrm{AF}\} \\ &= \{(R \circ \gamma')(x) \mid \gamma' \in \mathrm{AF}\} \\ &= \{\gamma(x) \mid \gamma \in \mathrm{AF_R}\} \end{aligned}$$

wobei mit AF die Menge der *Auswahlfunktionen*, d.h. die Menge der Funktionen, die Mengen auf eines ihrer Elemente abbilden, gemeint ist. Konsequenterweise ist dann $\mathrm{AF_R} := \{R \circ \gamma' \mid \gamma' \in \mathrm{AF}\}$. $\mathrm{AF_R}$ ist eine Menge von *Erreichbarkeitsfunktionen*, d.h. Funktionen, die Zustände auf über R erreichbare Zustände abbilden. Die Menge aller Argument-Wertepaare aller Funktionen in $\mathrm{AF_R}$ ergibt genau die Relation R ($\mathrm{AF_R}$ ist jedoch nicht die einzige Menge von Funktionen, mit der man R rekonstruieren kann). Das folgende Beispiel (ohne Parameter) soll dies verdeutlichen:



Relationale Repräsentation:

$R(z_1,z_2)$, $R(z_1,z_3)$, $R(z_1,z_4)$,
$R(z_2,z_5)$, $R(z_2,z_6)$

Funktionale Repräsentation:

Wir brauchen mindestens drei Funktionen:

$\gamma_1 : z_1 \mapsto z_2 \quad \gamma_2: \ z_1 \mapsto z_3 \quad \gamma_3: z_1 \mapsto z_4$
$\qquad z_2 \mapsto z_5 \qquad\quad z_2 \mapsto z_6$

Diese Menge von Erreichbarkeitsfunktionen ist minimal (aber nicht eindeutig). $\mathrm{AF_R}$ ist dagegen eine maximale Menge von Erreichbarkeitsfunktionen.

Leider sind die Zusammenhänge nur dann so einfach wie geschildert, wenn die Relation R *serial* ist, d.h. wenn $R(x) \neq \emptyset$ für alle x. Ist dies nicht der Fall, sind die Funktionen in $\mathrm{AF_R}$ partiell. Dies stellt im Prinzip kein wesentliches Problem dar, macht die Sache aber technisch komplizierter [Ohl88, Ohl91]. Daher beschränken wir uns im folgenden auf den serialen Fall.

Die Beziehung $R(x) = \{\gamma(x) \mid \gamma \in \mathrm{AF_R}\}$ erlaubt nun, die Semantikdefinition der Modaloperatoren in Def. 1 umzuschreiben

$$\begin{aligned} &\Im \models \Box\Phi \ \text{ gdw. für alle } u \in \mathrm{AF_R} \text{ gilt } \Im[u(z)] \models \Phi \\ &\Im \models \Diamond\Phi \ \text{ gdw. es gibt ein } u \in \mathrm{AF_R} \text{ und } \Im[u(z)] \models \Phi. \end{aligned}$$

Der wesentliche Unterschied zur Originaldefinition ist, daß die Bedingung „für alle z' mit $R(z,z')$“, die vom aktuellen Zustand z abhängt, durch „für alle $u \in \mathrm{AF_R}$“ ersetzt wurde, was einem ganz normalen Allquantor entspricht. Dies erlaubt nun, den $\Box$-Operator wirklich wie einen Allquantor zu behandeln. Die Bedingung $u \in AF_R$ läßt sich syntaktisch durch eine entsprechende *Typisierung* der Variablen u in einer geeigneten Sortenlogik ausdrücken (z.B. als $u{:}AF_R$).

Mit Hilfe dieser modifizierten Semantik wird die Übersetzungsfunktion Π_R (Def. 3) in Π_F abgewandelt, die z.B. die Formel $\Box\Diamond P$ übersetzt in $\exists a\ \forall u{:}AF_R\ \ \exists v{:}AF_R\ P'(v(u(a)))$, oder, wenn man $v(u(a))$ in Form von Funktionskompositionen schreibt in $\exists a\ \forall u{:}AF_R\ \exists v{:}AF_R\ P'((u \circ v)(a)))$.

Das führende $\exists a$ taucht immer in genau dieser Form auf. Daher kann man es auch weglassen und dann einfach schreiben: $\forall u{:}AF_R\ \ \exists v{:}AF_R\ P'([u\ v])$, wobei $[u\ v]$ für $(u \circ v)(a)$ steht. Auf komplexe Formeln angewendet, ergibt das eine wesentlich kompaktere Form, als die von Π_R produzierte.
Die Erweiterung von Π_F auf parametrisierte Modaloperatoren ist jetzt relativ einfach. Es gilt

$$\begin{aligned} R_p(x)\} &= \{y \mid R_p(x,y)\} \qquad\qquad (= \{y \mid y \in R_p(x)\}) \\ &= \{\gamma'(R_p(x)) \mid \gamma' \in AF\} \\ &= \{(R_p \circ \gamma')(x) \mid \gamma' \in AF\} \\ &= \{\gamma(p)(x) \mid \gamma \in AF'_R\}. \end{aligned}$$

AF'_R ist jetzt die Menge der parametrisierten Erreichbarkeitsfunktionen, d.h. für $\gamma \in AF'_R$ bildet $\gamma(p)$ auf Zustände ab, die über die p-parametrisierte Relation erreichbar sind.

Die entsprechende Semantikdefinition für die parametrisierten Modaloperatoren sieht jetzt folgendermaßen aus:

$\Im \models [p]\Phi$ gdw. für alle $u \in AF'_R$: $\Im[u(p)(z)] \models \Phi$
$\Im \models \langle p\rangle\Phi$ gdw. es gibt ein $u \in AF'_R$ mit $\Im[u(p)(z)] \models \Phi$.

(Man könnte natürlich anstelle von $u(p)(z)$ auch $u(p,z)$ schreiben. Die erste Version ist die sogenannte „curried" Version der zweiten und hat manchmal Vorteile.)

Wir definieren jetzt die funktionale Übersetzungsfunktion Π_F folgendermaßen.

Definition 22 (Funktionale Übersetzung in PL1)
Für eine modallogische Formel Φ sei $\Pi_F(\Phi) = \exists a\ \pi(\Phi,z)$, wobei $\pi(\Phi,z)$ folgendermaßen arbeitet (wiederum verwenden wir π auch, um Terne zu übersetzen):

$\pi(x,z) = x$ falls x eine Variable ist
$\pi(f(t_1,\ldots,t_n),z) = f'(z,\pi(t_1,z),\ldots,\pi(t_n,z))$ falls f ein Funktionssymbol ist
$\pi(P(t_1,\ldots,t_n),z) = exists(z, \pi(t_1,z)) \wedge \ldots \wedge exists(z, \pi(t_n,z)) \wedge P(z,\pi(t_1,z),\ldots, \pi(t_n,z))$
falls P ein Prädikatensymbol ist
$\pi(\neg\Phi, z) = \neg\pi(\Phi, z)$
$\pi(\Phi \wedge \Psi, z) = \pi(\Phi, z) \wedge \pi(\Psi, z)$
$\pi(\Phi \vee \Psi, z) = \pi(\Phi, z) \vee \pi(\Psi, z)$
$\pi(\forall x\ \Phi, z) = \forall x\ exists(z, x) \Rightarrow \pi(\Phi, z)$
$\pi(\exists x\ \Phi, z) = \exists x\ exists(z, x) \wedge \pi(\Phi, z)$
$\pi([p]\Phi, z) = exists(z, \pi(p,z)) \Rightarrow \forall u{:}AF_R\ \pi(\Phi, u(p)(z))$
$\pi(\langle p\rangle\Phi, z) = exists(z, \pi(p,z)) \wedge \exists u{:}AF_R\ \pi(\Phi, u(p)(z))$ ■

Der „exists(z, π(p,z)) …" Teil bei der Übersetzung der parametrisierten Modaloperatoren trägt der Möglichkeit Rechnung, daß die Parameter selbst Terme sind, welche auf Objekte referenzieren, die eventuell in bestimmten Zuständen überhaupt nicht mehr existieren. Dieser Teil fällt dann weg, wenn man unparametrisierte Modaloperatoren hat, oder wenn man von vorneherein weiß, daß die Parameter der Operatoren nur Objekte bezeichnen, die in allen Zuständen existieren.

Um die Syntax etwas zu vereinfachen, kann man, wie oben schon angedeutet, den äußeren Existenzquantor weglassen und die Zustandsterme als Strings schreiben. Nach der Übersetzung können in den Formeln Terme der Art x(p) auftauchen, wobei x eine Variable ist. Das sieht auf den ersten Blick aus wie ein Term aus Prädikatenlogik zweiter Ordnung. Mit einem einfachen Trick kann man das in *diesem* Fall jedoch wieder als Term in PL1 darstellen. Anstelle x(p) schreibt man apply(x,p) mit der intendierten Bedeutung von apply: Wende x auf p an. (Funktionsvariablen werden erst dann wirklich höherer Ordnung wenn man damit über *alle* Funktionen auf einer Menge quantifiziert, über die gleichzeitig auch ein normaler Quantor quantifiziert, also z.B. $\forall x{:}A\ \forall y{:}A \rightarrow A\ P(y(x))$ und man damit alle Funktionen von A nach A meint. Dieses *alle* geht verloren, man wenn man $\forall x\ \forall y\ P(apply(y,x))$ schreibt.)

Für die funktionale Übersetzung kann man jetzt ebenfalls Vollständigkeit und Korrektheit beweisen:

Eine modallogische Formel ist modallogisch erfüllbar gdw.
die übersetzte Formel prädikatenlogisch erfüllbar ist.

Mit Hilfe der funktionalen Übersetzung lassen sich jetzt weitere interessante Beziehungen zwischen modallogischen Axiomenschemata und Eigenschaften der semantischen Struktur herleiten. Dazu wenden wir wieder den SCAN-Algorithmus (Def. 6) an. Die Parameter der Modaloperatoren sollen bei diesen Beispielen immer persistent sein, d.h. nicht von den Zuständen abhängen. Bei der Übersetzung werden sie daher nicht verändert.

Als erstes Beispiel betrachten wir das Hierarchieaxiom $[p]\Phi \Rightarrow [q]\Phi$. Bezeichnen beispielsweise p und q Aktionen, dann besagt dieses Axiom: „Gilt nach Ausführung der Aktion p die Aussage Φ, so auch nach Ausführung der Aktion q“. Das ist nichts anderes als die Spezialisierungsbeziehung zwischen Aktionen, wie sie z.B. zwischen den Aktionen *bewegen* und *gehen* besteht.

Beispiel 23	**(Hierarchieaxiom)**
Axiomenschema	$[p]\Phi \Rightarrow [q]\Phi$
negiert:	$[p]\Phi \wedge \langle q\rangle\neg\Phi$
übersetzt:	$\exists a\ (\forall x\ P(x(p)(a))) \wedge (\exists x\ \neg P(x(q)(a)))$
Klauselform:	$P(x(p)(a))$
	$\neg P(c(q)(a))$
P wegresolviert:	$x(p)(a) \neq c(q)(a)$
entskolemisiert:	$\exists a,c\ \forall x\ x(p)(a) \neq c(q)(a)$
negiert:	$\forall x,y\ \ \exists z\ z(p)(x) = y(q)(x)$ ■

Mit einer entsprechend erweiterten Version von Π_R würde der SCAN-Algorithmus liefern:

$\forall x,y\ \ R_q(x,y) \Rightarrow R_p(x,y)$, d.h. R_q ist eine Teilrelation von R_p. Und in der Tat ist, wie man folgendermaßen sieht, $\forall x,y\ \exists z\ z(p)(a) = y(q)(a)$ das funktionale Äquivalent dieser Teilrelationsbeziehung:

$\forall x,y\ R_q(x,y) \Rightarrow R_p(x,y)$

gdw. $\forall x\ R_q(x) \subseteq R_p(x)$

gdw. $\forall x\ \forall y \in \mathrm{AF}\ \exists z \in \mathrm{AF}\ y(R_q(x)) = z(R_p(x))$

gdw. $\forall x\ \forall y \in \mathrm{AF}\ \exists z \in \mathrm{AF}\ (R_q \circ y)(x)) = (R_p \circ z)(x))$

gdw. $\forall x\ \forall y \in \mathrm{AF}'_R\ \exists z \in \mathrm{AF}\ y(q)(x)) = z(p)(x))$

Sind die Parameter p und q einfache Konstantensymbole, kann man die festgestellte Teilmengenbeziehung verwenden, um p und q nach der Übersetzung selbst als Sorten zu interpretieren und dann die Zustandsvariablen unter der Annahme der Untersortenbeziehung $q \sqsubseteq p$ zu unifizieren.

Das nächste Beispiel ist das Persistenzaxiom $[p]\Phi \Rightarrow [q][p]\Phi$.
(Ist p = q dann entspricht das dem Transitivitätsaxiom $\Box\Phi \Rightarrow \Box\Box\Phi$.)

Das Axiom findet z.B. Verwendung, wenn man ausdrücken möchte, daß ein Agent *A* nichts vergißt. Dann wählt man als Parameter p = weiß(A), und als Parameter q = immer und erhält dann [weiß(A)]Φ ⇒ [immer][weiß(A)]Φ, d.h. wenn er einmal Φ weiß, dann weiß er es immer. Dieses Beispiel illustriert eine sehr interessante Eigenschaft der parametrisierten Modaloperatoren: Man kann verschiedene Parameter gemischt betrachten, z.B. einmal als Zeitparameter, einmal als epistemische Parameter, usw. Der Kombination von Logiken ist damit fast keine Grenzen gesetzt.

Beispiel 24	**(Persistenzaxiom)**
Axiomenschema	$[p]\Phi \Rightarrow [q][p]\Phi$
negiert:	$[p]\Phi \wedge \langle q\rangle \langle p\rangle\neg\Phi$
übersetzt:	$\exists a\ (\forall x\ P(x(p)(a))) \wedge (\exists u\ \exists v\ \neg P((u(q) \circ v(p))(a)))$
Klauselform:	$P(x(p)(a))$
	$\neg\ P((c(q)\circ\ d(p))(a))$
P wegresolviert:	$x(p)(a) \neq (c(q)\circ\ d(p))(a)$
entskolemisiert:	$\exists a,c,d\ \forall x\ x(p)(a) \neq (c(q)\circ\ d(p))(a)$
negiert:	$\forall x,y,z\ \ \exists v\ \ v(p)(x) = (y(q)\circ\ z(p))(x)$ ■

Mit dem nächsten Beispiel wird illustriert, wie man den Mechanismus verwenden kann, um Operationen auf den Parametern zu definieren. Angenommen, wir haben eine Aktionslogik, in der [p]P bedeutet: Nach Ausführung von Aktion p gilt Φ. Jetzt wollen wir, wie in dynamischer Logik, eine Operation “Hintereinanderausführung von Aktionen“ einführen. [p;q]Φ soll heißen: Nachdem die Operation *p;q*, die aus der Hintereinanderausführung von *p* und *q* besteht, ausgeführt wurde, gilt Φ. Hierfür gilt offensichtlich das Axiomenschema $[p][q]\Phi \Rightarrow [p;q]\Phi$. Auf dieses Schema wenden wir den SCAN-Algorithmus an.

Beispiel 25	**(Sequenzierung)**
Axiomenschema	$[p][q]\Phi \Rightarrow [p;q]\Phi$

negiert:	$[p][q]\Phi \wedge \langle p;q\rangle\neg\Phi$
übersetzt:	$\exists a\ (\forall x,v\ P((u(p) \circ v(q))(a)) \wedge (\exists c\ \neg P(c(p;q)(a))$
Klauselform:	$P((u(p) \circ v(q))(a))$ $\neg\ P(c(p;q)(a))$
P wegresolviert:	$(u(p) \circ v(q))(a) \neq c(p;q)(a)$
entskolemisiert:	$\exists a,c\ \forall u,v\ (u(p) \circ v(q))(a) \neq c(p;q)(a)$
negiert:	$\forall x,y\ \ \exists u,v\ (u(p) \circ v(q))(x) = y(p;q)(x)$

Dieses Axiom beschreibt eine Art Dichtheit der semantischen Struktur: Zwischen jeweils zwei Zuständen für p und q liegt derjenige für p;q noch dazwischen. ■

Mit der funktionalen Übersetzung lohnt es sich, noch einmal das McKinsey Axiomenschema (Beispiel 19) zu untersuchen.

Beispiel 26	**(McKinsey's Axiom)**
Axiomenschema	$\Box\Diamond\Phi \Rightarrow \Diamond\Box\Phi$
negiert:	$\Box\Diamond\Phi \wedge \Box\Diamond\neg\Phi$
übersetzt:	$\exists a\ \forall u\ \ \exists v\ P((u \circ v)(a)) \wedge\forall\ u\ \ \exists v\ \neg P((u \circ v)(a))$
Klauselform:	$P((u \circ g(u))(a))$ $\neg P((u' \circ\ h(u'))(a))$
P wegresolviert:	$(u \circ g(u))(a) \neq (u' \circ h(u'))(a)$
entskolemisiert:	$\exists a\ \begin{matrix}\exists g\ \forall u\\ \exists h\ \forall u'\end{matrix}\ (u \circ\ g(u))(a)) \neq (u' \circ\ h(u'))(a)$
negiert:	$\forall x\ \begin{matrix}\forall g\ \exists u\\ \forall h\ \exists u'\end{matrix}\ (u \circ\ g(u))(x)) \neq (u' \circ\ h(u'))(x)$

Dabei wurde ein paralleler Henkin-Quantor, sowie Quantoren über Funktionen benötigt. Skolemisiert man die erzeugte Formel wieder, dann ergibt sich:

$$\forall x\ \forall g,h\ \ (a(x,g) \circ g(a(x,g)))(x)) = (b(x,h) \circ h(b(x,h)))(x) \qquad ■$$

Dies ist eine „relativ harmlose“ Formel 2. Ordnung, mit der man durchaus noch umgehen kann. Der Leser möge z.B. versuchen, die Formel $\Box\Box\Diamond\ P \Rightarrow \Diamond\Diamond\Box P$ mit Hilfe der funktionalen Übersetzungsmethode und des gewonnen Korrespondenzaxioms zu beweisen.

Offensichtlich produziert der SCAN-Algorithmus mit der funktionalen Übersetzung vorwiegend Gleichungen. Da diese Gleichungen charakteristisch für eine ganze Logik sind, und deshalb für diese Logik immer benötigt werden, bietet es sich an, solche Gleichungen in einen Theorieunifikationsalgorithmus einzubauen. Wenn das möglich ist, dann ist das die effizienteste Art und Weise, diese Logik zu automatisieren. In [Ohl88,Ohl91] sind Unifikationsalgorithmen angegeben für die Fälle $\Box\Phi \Rightarrow \Phi$ (Reflexivität), $\Box\Phi \Rightarrow \Box\Box\Phi$, (Transitivität) und $\Phi \Rightarrow \Box\Diamond\Phi$ (Symmetrie) sowie deren beliebige Kombinationen.

Zum Ende dieses Abschnitts wird zur Illustration des vorgestellten Apparats ein logisches Rätsel behandelt, welches von McCarthy in die Diskussion gebracht wurde. Logiker und

Philosophen benutzen es manchmal, um die Adäquatheit ihrer Logik zu demonstrieren. Es gibt verschiedene Varianten davon. Hier benutzen wir eine, die zwar unfair, dafür aber logisch einwandfrei ist.

Beispiel 27 (Wise Men Puzzle)

> *Es war einmal ein König, der hatte drei weise Männer. Eines Tages beschloß er, herauszufinden, wer von den dreien der weiseste ist. Er stellt sie einander gegenüber, macht jedem von ihnen einen weißen oder schwarzen Punkt auf die Stirn und sagt ihnen, daß mindestens einer der Punkte weiß ist. Nach einer Weile, in der niemand etwas sagte, fragte er denjenigen, den er sowieso für den dümmsten hielt, ob er die Farbe seines Punktes wüßte. Der verneinte. Dann fragte er den vermeintlich zweitdümmsten. Der verneinte ebenfalls. Daraufhin behauptete der dritte prompt, sein Punkt sei weiß. Woher konnte er das wissen?*[10]

Zur Lösung dieses Rätsels ist es notwendig, Schlüsse über den Wissensstand der Beteiligten zu ziehen. Zur Formulierung dieses Rätsels ist die prädikatenlogische Sprache daher nicht sehr gut geeignet. Viel einfacher wird es in epistemischer Logik.

Neben dem Gleichheitssymbol $=$ und dem Ungleichheitssymbol $\neq$ führen wir folgende syntaktische Einheiten ein: A, B und C sind Konstantensymbole, die die drei weisen Männer bezeichnen[11]. *C* ist derjenige der die Antwort wußte. $\mathsf{w(x)}$ soll heißen: *x* hat einen weißen Punkt. Der parametrisierte Modaloperator $[\mathsf{p}]$ ist jetzt zu interpretieren als „*p* weiß, daß…“, während das Gegenstück $\langle\mathsf{p}\rangle$ bedeutet: „*p* hält für möglich, daß…“.

Außerdem benutzen wir, diesmal quasi als Abkürzung, den normalen $\Box$-Operator. Er steht für beliebig geschachtelte Sequenzen von $[\mathsf{S}]$-Operatoren für alle beteiligten Weisen *S*. Gäbe es nur die beiden A und B, dann stünde $\Box\Phi$ für $[\mathsf{A}]\Phi \wedge [\mathsf{B}]\Phi \wedge [\mathsf{A}][\mathsf{B}]\Phi \wedge [\mathsf{B}][\mathsf{A}]\Phi \wedge [\mathsf{A}][\mathsf{A}]\Phi \wedge [\mathsf{B}][\mathsf{B}]\Phi \wedge [\mathsf{A}][\mathsf{B}][\mathsf{A}]\Phi \wedge \ldots$, d.h. *A* weiß Φ und *B* weiß Φ und *A* weiß, daß *B* Φ weiß, und *B* weiß, daß *A* Φ weiß, und *A* weiß, daß er selbst Φ weiß, und *B* weiß, daß er selbst Φ weiß, und so weiter, ad infinitum. In unserem Fall mit drei Weisen, erweitert sich das entsprechend. $\Box\Phi$ bedeutet damit so etwas wie: „Φ ist allgemein bekannt“.

Die Information, die notwendig ist, um das Rätsel zu lösen, läßt sich jetzt in folgende Axiome kodieren:

$\mathsf{B} \neq \mathsf{A} \wedge \mathsf{C} \neq \mathsf{A} \wedge \mathsf{C} \neq \mathsf{B}$	(alle drei Weisen sind verschieden)
$\Box(\mathsf{w(A)} \vee \mathsf{w(B)} \vee \mathsf{w(C)})$	(einer hat einen weißen Punkt und das ist allgemein bekannt)
$\Box(\forall \mathsf{x,y}\ \mathsf{x} \neq \mathsf{y} \Rightarrow (\neg \mathsf{w(x)} \Rightarrow [\mathsf{y}]\neg \mathsf{w(x)}))$	(es ist allgemein bekannt, daß, wenn jemand keinen weißen Punkt hat, dies die anderen wissen.)
$[\mathsf{C}]\ [\mathsf{B}]\ \neg[\mathsf{A}]\ \mathsf{w(A)}$	(*C* weiß, daß *B* weiß, daß *A* die Farbe seines Punktes nicht kennt)

---

[10]Unfair an dieser Version ist, daß der dritte der Männer ausnutzen kann, daß die beiden anderen nichts wissen. Ohne diese Information können die beiden anderen prinzipiell nicht herausbekommen, welche Farbe ihr Punkt hat.

[11]Zur Unterscheidung zwischen Syntax und Semantik wird wieder die Konvention benutzt, daß z.B. das Symbol A den Mann *Ali*, das Symbol B den Mann *Jussuf* und das Symbol C den Mann *Hassan* bezeichnet. Um die Zuordnung zu erleichtern, benutzen wir aber nicht solche Namen, sondern schreiben *A*, *B* und *C* wenn die Männer gemeint sind.

$[C]\neg[B]\, w(B)$ (*C* weiß, daß *B* die Farbe seines Punktes nicht kennt)

Das zu beweisende Theorem ist: $[C]w(C)$, d.h. *C* weiß, daß er einen weißen Punkt hat.

Mit der funktionalen Übersetzung (Def. 22) können jetzt die Formeln des Beispiels übersetzt und damit die Operatoren eliminiert werden.

$$B \neq A \wedge C \neq A \wedge C \neq B$$
$$\forall u\, (w([u], A) \vee w([u], B) \vee w([u], C))$$
$$\forall v\, (\forall x,y\;\; x \neq y \Rightarrow (\neg w([v], x) \Rightarrow \forall w_y\, \neg w([w_y], x))$$
$$\forall u_C\, \forall v_B\, \neg \forall x_A\, w([u_C\, v_B\, x_A], A)$$
$$\forall u_C\, \neg \forall z_B\, w([u_C\, z_B], B)$$

Theorem: $\forall u_C\, w([u_C], C)$

(Eigentlich sollte man für jedes Vorkommen eines Quantors einen anderen Variablennamen wählen. Das erschwert aber manchmal die Lesbarkeit.)

Die Formeln werden in Klauselnormalform transformiert und dann das Theorem mit Resolution abgeleitet. (Wir machen einen positiven Beweis und keinen Widerlegungsbeweis.)

Die Klauselform lautet:

C_1:	$B \neq A$	
C_2:	$C \neq A$	
C_3:	$C \neq B$	
C_4:	$w([u], A) \vee w([u], B) \vee w([u], C)$	
C_5:	$x = y \vee w([v], x) \vee \neg w([v\, w_y], x)$	
C_6:	$\neg w([u_C\, v_B\, f_A(u_C, v_B)], A)$	(f_A stammt von $\neg\forall\, x_A \ldots = \exists x_A \neg \ldots$)
C_7:	$\neg w([u_C\, g_B(u_C)], B)$	(g_A stammt von $\neg\forall\, z_B \ldots = \exists z_B\, \neg \ldots$)

Für die Resolutionsableitung wird ausgenutzt, daß der □-Operator diesesmal für beliebige Sequenzen von parametrisierten □-Operatoren steht. Daher kann man die Variablen, die aus der Übersetzung des □-Operators stammen, mit beliebigen Sequenzen von aus der Übersetzung der anderen Operatoren stammenden Termen unifizieren. Die Resolutionsableitung läuft dann folgendermaßen ab:

C_5,1 & C_1	$\to C_8$:	$w([v],B) \vee \neg\, w([v\, w_A], B)$	$x \mapsto B,\ y \mapsto A$
C_8,1 & C_7	$\to C_9$:	$\neg w([u_C\, g_B(u_C)\, w_A], B)$	$v \mapsto [u_C\, g_B(u_C)]$
C_4,1 & C_6	$\to C_{10}$:	$w([u_C\, v_B\, f_A(u_C, v_B)], B) \vee w([u_C\, v_B\, f_A(u_C, v_B)], C)$	$u \mapsto [u_C\, v_B\, f_A(u_C, v_B)]$
C_{10},1 & C_9	$\to C_{11}$:	$w([u_C\, g_B(u_C)\, f_A(u_C,g_B(u_C))], C)$	$v_B \mapsto g_B(u_C),\ w_A \mapsto f_A(u_C,g_B(u_C))$
C_5,1 & C_2	$\to C_{12}$:	$w([v], C) \vee \neg\, w([v\, w_A], C)$	$x \mapsto C,\ y \mapsto A$
C_{12},2 & C_{11}	$\to C_{13}$:	$w([u_C\, g_B(u_C)], C)$	$v \mapsto [u_C\, g_B(u_C)],\ w_A \mapsto f_A(u_C,g_B(u_C))$
C_5,1 & C_3	$\to C_{14}$:	$w([v], C) \vee \neg w([v\, w_B], C)$	$x \mapsto C,\ y \mapsto B$
C_{14},2 & C_{13}	$\to C_{15}$:	$w([u_C], C)$	$v \mapsto u_C,\ w_B \mapsto g_B(u_C)$

C_{15} entspricht gerade dem zu beweisende Theorem $[C]w(C)$. Der Beweis ist zunächst etwas unübersichtlich. Wenn man aber einige Zwischenresultate zurückübersetzt in die Schreibweise mit Modal-Operatoren, bekommt man ein besseres Gefühl dafür, wie der Beweis durchgeführt wird. Es ist nämlich

C_9:	$\neg w([u_C\ g_B(u_C)\ w_A], B)$	=	$[C]\langle B\rangle\ [A]\neg\ w(B) = [C]\neg\ [B]\ \langle A\ \rangle\ w(B)$
C_{11}:	$w([u_C\ g_B(u_C)\ f_A(u_C, g_B(u_C))], C)$	=	$[C]\langle B\rangle\ [A]\ w(C)$
C_{13}:	$w([u_C\ g_B(u_C)], C)$	=	$[C]\ \langle B\rangle\ w(C)$
C_{15}:	$w([u_C], C)$	=	$[C]w(C)$

Übertragen in Umgangssprache lautet der Beweis dann ungefähr: Da *C* weiß, daß *B* die Farbe seines Punktes nicht kennt ($[C]\neg[B]w(B)$), weiß *C*, daß *B* es für möglich hält, daß er keinen weißen Punkt hat ($[C]\langle B\rangle\neg w(B)$). Da *A B* sehen kann, und *C* das weiß, schließt nun *C*, daß *B* es für möglich hält, daß *A* weiß, daß *B* selbst keinen weißen Punkt hat ($[C]\langle B\rangle[A]\neg w(B)$). Da weiterhin *C* weiß, daß *B* weiß, daß *A* es für möglich hält, daß er selbst, d.h. *A*, keinen weißen Punkt hat ($[C]\ [B]\neg[A]\ w(A) = [C]\ [B]\ \langle A\rangle\ \neg w(A)$), andererseits aber mindestens einer einen weißen Punkt haben muß und jeder das weiß, kann *C* mit dem vorherigen Ergebnis ($[C]\langle B\rangle\ [A]\neg\ w(B)$) weiter schließen, daß *B* es für möglich hält, daß *A* weiß, daß *C* einen weißen Punkt hat ($[C]\langle B\rangle\ [A]\ w(C)$). Da *A* jedoch *C* sehen kann und daher genau weiß, ob *C* einen weißen Punkt hat oder nicht, und *B* das weiß, kann das nur der Fall sein, wenn auch *B* selbst es für möglich hält, daß *C* einen weißen Punkt hat ($[C]\ \langle B\rangle\ w(C)$). *B* selbst kann aber auch *C* sehen. Daher kann *C* schließlich folgern, daß auch dies nur sein kann, wenn *C* tatsächlich einen weißen Punkt hat. Also weiß er, daß sein Punkt weiß ist. ■

Dieses Beispiel soll demonstrieren, daß die Formalisierung eines Problems in einer adäquaten Logik zu einer kompakteren und einfacheren Darstellung führen kann und, daß auch die Schlußfolgerungen mit einer rein syntaktisch arbeitenden Standardregel wie Resolution eine Vielzahl an Information verarbeiten kann.

Erste Hinweise zur Konstruktion von Logiken:

Wir können nun ein erstes Fazit zum Thema Konstruktion und Automatisierung von Logiken ziehen.

Wird eine Logik benötigt, die auf Prädikatenlogik erster Stufe aufbaut und zusätzlich einstellige Operatoren haben soll, dann ist folgendes Vorgehen zu empfehlen:

- Die verschiedenen Operatoren werden als parametrisierte Modaloperatoren aufgefaßt.
- Die Eigenschaften der Operatoren und insbesondere deren Beziehungen untereinander werden als Axiomenschemata aufgeschrieben.
- Mit dem SCAN-Algorithmus wird die korrespondierende semantische Eigenschaft ermittelt.
- Wann immer möglich, wird die mit dem SCAN-Algorithmus ausgerechnete Formel in einen Theorieunifikations- oder -resolutionsalgorithmus umgewandelt. (Dieser Schritt läßt sich derzeit noch nicht voll automatisieren.)
- In konkreten Anwendungen werden die dort anfallenden Formeln in PL1 übersetzt und dann mit einem für PL1 entwickelten Inferenzsystem bearbeitet.

Dieses Vorgehen kann an verschiedenen Stellen fehlschlagen. Der erste kritische Punkt ist, daß in dem bisher vorgestellten Formalismus immer noch die Notwendigkeitsregel

$$\frac{\Phi}{[p]\Phi} \quad \Phi \text{ ist eine Tautologie}$$

automatisch gilt. Diese Regel kann für bestimmte Anwendungen völlig inadäquat sein. Will man z.B. einen Want-Operator einführen mit der Bedeutung $[\mathsf{Want}(A)]\Phi$ heißt Agent *A* will Φ erreichen, dann bewirkt die Notwendigkeitsregel u.a.

$$\frac{\mathsf{Zahn\text{-}ziehen} \Rightarrow \mathsf{Schmerzen}}{[\mathsf{Want}(A)](\mathsf{Zahn\text{-}ziehen} \Rightarrow \mathsf{Schmerzen})} \quad \text{(Tautologie in irgendeiner Hintergrundtheorie)}$$

d.h. Agent *A* will erreichen, daß Zähne ziehen Schmerzen verursacht. Das ist offensichtlich Unsinn. Im nächsten Abschnitt werden wir sehen, wie die Notwendigkeitsregel abgeschwächt werden kann.

Die zweite Stelle an der das Verfahren schief gehen kann, ist die mögliche Erzeugung der leeren Klauselmenge im SCAN Algorithmus. Das passiert z.B., wenn man einen Operator „sagt" einführen will mit der Bedeutung $[\mathsf{sagt}(A)]\Phi$ soll heißen: Agent *A* sagt Φ, und dann damit einen notorischen Lügner axiomatisieren will. Wie in Beispiel 16 gezeigt wurde, kollabiert das Lügneraxiom $[\mathsf{sagt}(A)]\ \Phi \Rightarrow \neg\Phi$ zur leeren Klauselmenge. Das bedeutet, daß die normale Semantik des $\Box$-Operators in diesem Fall nicht angemessen ist. An dieser Stelle muß man dann eingreifen und eine andere Semantik wählen.

Die letzte kritische Stelle für den SCAN-Algorithmus ist die Möglichkeit, daß dieser nicht terminiert oder die Entskolemisierung nicht durchgeführt werden kann. In diesem Fall befindet man sich i.allg. nicht mehr in Logik erster Stufe. Benötigt man das auf diese Weise nicht zu handhabende Axiomenschema in seiner vollen Stärke, kann man keinen vollständigen Kalkül mehr erwarten und muß sich mit anderen, bisher wenig untersuchten Methoden behelfen.

Ist die Anwendung jedoch so, daß nur Fragmente der vollen Logik, z.B. ohne Quantoren, gebraucht werden, so kann es sein, daß die Logik trotz höherer-Ordnung-Eigenschaften der Semantik noch entscheidbar (z.B. Logik G) oder zumindest semientscheidbar ist. In diesem Fall muß man entsprechende Spezialverfahren verwenden (die teilweise noch zu entwickeln sind).

1.7 Nachbarschaftssemantik

In diesem Abschnitt wird eine Methode vorgestellt, dem Problem der u.U. zu starken Notwendigkeitsregel Herr zu werden. Zur Erinnerung: In den bisher angegebenen Versionen der Semantik des $\Box$-Operators gilt $\Box\Phi$ für beliebige Tautologien Φ, d.h. $\Box\mathsf{wahr}$ ist ein Axiom der Logik. Wie schon am Beispiel des Want-Operators demonstriert, kann das für bestimmte Anwendungen inadäquat sein. Um die Notwendigkeitsregel vermeiden zu können, muß man die Semantik des $\Box$-Operators abschwächen.

Eine Möglichkeit hierzu bietet die sogenannte Nachbarschaftssemantik oder auch „Minimal Model" Semantik [Che80, Rau79]. Hierbei betrachtet man immer noch Frames mit Zuständen, aber die Erreichbarkeitsrelation wird ersetzt durch eine Nachbarschaftsfunktion $\mathbb{N}$. Für jeden

Zustand z berechnet $\mathbb{N}$ eine Menge von Zustandsmengen, den sogenannten Nachbarschaften von z. Eine Formel $\Box\Phi$ ist genau dann wahr in einem Zustand z wenn Φ *in genau* einer der von $\mathbb{N}$ berechneten Nachbarschaften wahr ist. Um den Wahrheitsgehalt von $\Box\Phi$ zu ermitteln, muß also eine Nachbarschaft N von z gefunden werden, die genau die Welten umfaßt, in denen Φ wahr ist. Die formalen Definitionen lauten dann wie folgt:

Definition 28 (Nachbarschaftssemantik)

Ein Frame $F = (Z, \mathbb{N})$ in Nachbarschaftssemantik besteht aus einer Menge Z von Zuständen sowie einer Funktion $\mathbb{N}: Z \to 2^{2^Z}$.

Interpretationen $\Im = (F, z, \Theta, V)$ sind analog zur relationalen Semantik (Def. 1) definiert.

Die Semantikregeln für die Modaloperatoren sehen folgendermaßen aus: Für eine Interpretation $\Im$ mit aktuellem Zustand z gilt:

$\Im \models \Box\Phi$ gdw. es existiert ein $N \in \mathbb{N}(z)$ so daß für alle z' gilt: $z' \in N$ gdw. $\Im[z'] \models \Phi$
$\Im \models \Diamond\Phi$ gdw. für alle $N \in \mathbb{N}(z)$ existiert ein z' mit: $z' \in N$ gdw. $\Im[z'] \models \Phi$

Alle anderen Definitionen ändern sich gegenüber denen in der relationalen Semantik nicht. ❑

Da *wahr* in allen Zuständen gilt, kann man sich jetzt leicht überlegen, daß $\Box$wahr nur noch dann gelten kann, wenn die Menge aller Zustände in der Nachbarschaft des aktuellen Zustands liegt, und dies ist nicht ausdrücklich verlangt. Also ist $\Box$wahr kein Axiom mehr, und die Notwendigkeitsregel ist außer Kraft gesetzt.

Eine Regel, die aber immer noch gilt, ist folgende:

$$\frac{\Phi \Leftrightarrow \Psi}{\Box\Phi \Leftrightarrow \Box\Psi} \qquad \Phi \Leftrightarrow \Psi \text{ ist eine Tautologie}$$

d.h. man kann innerhalb eines modalen Kontexts Formeln durch dazu äquivalente Formeln austauschen. Z.B. ist $\Box(\Phi \wedge \Psi)$ äquivalent zu $\Box(\Psi \wedge \Phi)$. Das sieht vernünftig aus, kann aber für manche Anwendungen immer noch zu stark sein.

Die Gültigkeit dieser Regel in der Nachbarschaftssemantik ist leicht einzusehen. Wenn Φ genau in den Zuständen einer bestimmten Nachbarschaft N des aktuellen Zustands gilt und sonst nirgends, und weiterhin Φ äquivalent ist zu Ψ (d.h. Φ in genau den Welten gilt, in denen auch Ψ wahr ist), dann gilt natürlich Ψ auch genau in den Zuständen von N und sonst nirgends. Also ist $\Box\Phi \Leftrightarrow \Box\Psi$ ein gültiges Axiomenschema.

Eine Regel, die nicht mehr automatisch gilt, ist die folgende:

$$\frac{\Phi \Rightarrow \Psi}{\Box\Phi \Rightarrow \Box\Psi} \qquad \Phi \Rightarrow \Psi \text{ ist eine Tautologie}$$

Auch das ist leicht einzusehen. Gilt Φ genau in den Zuständen einer bestimmten Nachbarschaft N des aktuellen Zustands und sonst nirgends, und ist $\Phi \Rightarrow \Psi$ wahr, so gilt zwar Ψ auch in den Zuständen von N, aber nicht notwendigerweise nur in diesen. Daher muß es nicht so sein, daß $\Box\Psi$ gilt.

Diese Regel kann man „einschalten“, indem man entweder in der Semantikdefinition des $\Box$-Operators die Bedingung

$$z' \in N \text{ gdw. } \Im[z'] \models \Phi \qquad \text{zu} \qquad z' \in N \text{ impliziert } \Im[z'] \models \Phi$$

abschwächt, oder aber fordert, daß alle von $\mathbb{N}$ berechneten Nachbarschaftsmengen nach oben abgeschlossen sind. Das heißt, wenn $N \in \mathbb{N}(z)$, dann sind alle Obermengen von N ebenfalls in $\mathbb{N}(z)$.

Als nächstes interessiert uns wieder die Übersetzung in Prädikatenlogik. Dazu schreiben wir die Semantikdefinition zunächst im Sinne der funktionalen Übersetzung um.

Es ist
$\Im \models \Box\Phi$

gdw. es existiert ein $N'' \in \mathbb{N}(z)$, so daß für alle z' gilt: $z' \in N''$ gdw. $\Im[z'] \models \Phi$
gdw. es existiert ein $N' \in$ AF, so daß für alle z' gilt: $z' \in N'(\mathbb{N}(z))$ gdw. $\Im[z'] \models \Phi$
gdw. es existiert ein $N \in AF_N$, so daß für alle z' gilt: $z' \in N(z)$ gdw. $\Im[z'] \models \Phi$

Die Semantik des $\Diamond$-Operators ist dann analog:

$$\Im \models \Diamond\Phi \quad \text{gdw. für alle } N \in AF_N \text{ existiert ein } z' \text{ mit: } z' \in N(z) \text{ gdw. } \Im[z'] \models \Phi$$

Dabei ist wieder AF die Menge der Auswahlfunktionen und

$$AF_N = \{\mathbb{N} \circ \gamma \mid \gamma \in AF\}$$

die Menge der Funktionen, die für beliebige Zustände jeweils eine bestimmte Nachbarschaft auswählen. Mit dieser vereinfachten Semantik kann man jetzt wieder eine Übersetzungsfunktion Π_N in Prädikatenlogik definieren:

Definition 29 („Nachbarschafts“-Übersetzung in PL1)
Für eine modallogische Formel Φ sei $\Pi_N(\Phi) = \exists a\, \pi(\Phi,a)$, wobei $\pi(\Phi,z)$ wie für Π_F (Def. 22) arbeitet, wobei nur die folgenden Änderungen zu betrachten sind:

$$\pi(\Box\Phi, z) = \exists X\, \forall x\; x \in X(z) \Leftrightarrow \pi(\Phi,x)$$
$$\pi(\Diamond\Phi, z) = \forall X\, \exists x\; x \in X(z) \Leftrightarrow \pi(\Phi,x) \qquad \blacksquare$$

Die neue Übersetzungsfunktion Π_N verwenden wir jetzt, um mit Hilfe des SCAN-Algorithmus (Def. 6) Korrespondenzen zwischen den bekannten Axiomenschemata und Eigenschaften der Nachbarschaftsstruktur zu finden. Dazu beginnen wir mit dem bekannten Schema $\Box\Phi \Rightarrow \Phi$.

Beispiel 30

Axiomenschema	$\Box\Phi \Rightarrow \Phi$
negiert:	$\Box\Phi \wedge \neg\Phi$
übersetzt:	$\exists a\, (\exists X\, \forall x\; x \in X(a) \Leftrightarrow P(x)) \wedge \neg P(a)$
Klauselform:	$x \notin A(a), P(x)$
	$x \in A(a), \neg P(x)$
	$\neg P(a)$
P wegresolviert:	$a \notin A(a)$

entskolemisiert:	$\exists a,A\ a \notin A(a)$
negiert:	$\forall x,X\ x \in X(x)$

Mit anderen Worten, $\Box\Phi \Rightarrow \Phi$ korrespondiert zu der Eigenschaft der Nachbarschaftsstruktur, daß jeder Zustand in jedem seiner Nachbarschaften enthalten ist. ■

Analog kann man zeigen, daß das Lügneraxiom $\Box\Phi \Rightarrow \neg\Phi$ besagt, daß kein Zustand in keinem seiner Nachbarschaften enthalten ist.

Dieses Axiomenschema besitzt eine noch relativ einfache korrespondierende Eigenschaft. Um zu zeigen, daß häufig wesentlich kompliziertere Korrespondenzen vorkommen, untersuchen wir das Schema $\Box\Phi \Rightarrow \Box\Box\Phi$, welches in der relationalen Semantik der Transitivität der Erreichbarkeitsrelation entspricht.

Um das Verfahren etwas abzukürzen, erlauben wir uns an dieser Stelle die Umrechnung in Klauselform zu überspringen.

Beispiel 31

Axiomenschema	$\Box\Phi \Rightarrow \Box\Box\Phi$
negiert:	$\Box\Phi \wedge \Diamond\Diamond\neg\Phi$
übersetzt:	$\exists a\ (\exists X\ \forall x\ x \in X(a) \Leftrightarrow P(x)) \wedge$ $\forall X\ \exists x\ x \in X(a) \Leftrightarrow (\forall Y\ \exists y\ y \in Y(x) \Leftrightarrow \neg P(y))$
transformiert:	Da der erste Teil aus einer Äquivalenz besteht, läßt sie sich direkt in den zweiten Teil einsetzen, um damit das P-Literal zu eliminieren. Wir erhalten: $\exists a,X,x\ x \in X(a) \Leftrightarrow (\forall Y\ \exists y\ y \in Y(x) \Leftrightarrow y \notin X(a))$
negiert:	$\forall\ a,X,x\ x \in X(a) \Leftrightarrow (\exists Y\ Y(x) = X(a))$ ■

Das Beispiel verdeutlicht, daß der Preis für die erweiterten Möglichkeiten der abgeschwächten Semantiken aus der größeren Komplexität der Behandlung schon bekannter Formeln besteht. (Das Phänomen dürfte jedem aus dem täglichen Leben bekannt sein: Je mehr Knöpfe ein Gerät hat, an denen man drehen kann, umso komplizierter wird es, auch nur einfache Dinge damit zu machen.)

Die Erweiterung der Idee der Nachbarschaftssemantik auf parametrisierte Operatoren ist jetzt kein Problem mehr und bleibt dem Leser als Übungsaufgabe überlassen.

2 Temporallogik

Eine der naheliegendsten Interpretationen der in der Semantik der Modallogik verwendeteten Zustände ist diese: Zustände entsprechen Zeitpunkten, und die Erreichbarkeitsrelation modelliert den Fluß der Zeit. Mit dieser Sichtweise lassen sich die Modaloperatoren $\Box$ als *immer* und $\Diamond$ als *irgendwann* interpretieren. Der aktuelle Zustand z in einer Interpretation $\mathfrak{I}$ (Def. 1) läßt sich dann als „jetzt“ deuten [Ben83]. Mit diesen beiden Begriffen kann man aber nur relativ wenige zeitliche Bezüge ausdrücken. Daher muß man die Logik noch erheblich verfeinern und ausbauen.

In diesem Abschnitt sollen die Ideen und Methoden, die im Abschnitt über Modallogiken dargestellt wurden, angewendet werden, um verbesserte Versionen von Zeitlogiken — bekannte sowie neue — zu definieren. Da wir zur Mechanisierung die Übersetzung in Prädikatenlogik, relational (Def. 3) sowie funktional (Def. 22), vorschlagen, können problemlos beliebige Schachtelungen von Zeitoperatoren und normalen Quantoren zugelassen werden. Dieser Abschnitt ist in erster Linie als Fallstudie zur *Konstruktion von Logiken* zu sehen.

2.1 Konstruktion der Grundlogik

Bevor wir mit der Konstruktion der Logik anfangen, müssen wir uns über den adäquaten Semantikformalismus klar werden, d.h. wir müssen uns entscheiden zwischen relationaler Semantik (Def. 1) oder Nachbarschaftssemantik (Def. 28). Das entscheidende Kriterium ist, ob wir die Notwendigkeitsregel

$$\frac{\Phi}{\Box\Phi} \qquad \Phi \text{ ist eine Tautologie}$$

akzeptieren können oder nicht, oder mit anderen Worten, ob $\Box$wahr gelten soll oder nicht. Unter der intendierten Interpretation von $\Box$ als *immer* ist das offensichtlich genau das Richtige — Tautologien sind eben *immer* wahr. Also wählen wir in diesem Fall die relationale Semantik. Damit haben wir die Erreichbarkeitsrelation zur Verfügung und können sie so spezifizieren, daß sie unserer Vorstellung vom Fluß der Zeit entspricht.

Mit den einfachen $\Box$ und $\Diamond$-Operatoren kann man nicht zwischen Zukunft und Vergangenheit unterscheiden. Um diese Unterscheidung machen zu können, werden parametrisierte Operatoren eingeführt. Zunächst führen wir folgende drei Parameter ein mit den intendierten Bedeutungen:

Definition 32 (Grundlegende Zeitoperatoren)

$[\mathsf{F}]\Phi$	Φ gilt *immer in der Zukunft*	(Future)
$[\mathsf{P}]\Phi$	Φ gilt *immer in der Vergangenheit*	(Past)
$[\mathsf{A}]\Phi$	Φ gilt *immer*	(Always)

Dazu gehören dann noch die dualen Operatoren

$\langle\mathsf{F}\rangle\Phi$	Φ gilt *irgendwann in der Zukunft*
$\langle\mathsf{P}\rangle\Phi$	Φ gilt *irgendwann in der Vergangenheit*
$\langle\mathsf{A}\rangle\Phi$	Φ gilt *irgendwann* .

Entsprechend der relationalen Semantik gehören dazu die drei Erreichbarkeitsrelationen R_F, R_P und R_A. ■

Als nächstes muß festgelegt werden, welche grundlegenden Eigenschaften diese Relationen haben, und welche Korrelationen es zwischen ihnen gibt. Wie wir im Abschnitt über Modallogiken gesehen haben, äußert sich das auf drei Ebenen: Zunächst betrachten wir die Eigenschaften der Relationen selbst. In der relationalen Übersetzung in Prädikatenlogik (Def. 3) würde man diese Eigenschaften unmittelbar als Axiome über den Prädikatensymbolen R_F, R_P

und R_A ausdrücken. Auf der Formelebene entsprechen die Eigenschaften der Relationen charakteristischen Axiomenschemata, und schließlich erhält man mit der funktionalen Übersetzung entsprechende Axiome in Termini der Erreichbarkeitsfunktionen AF_R. Im folgenden werden alle drei Versionen angegeben. Da wir nur eine feste Menge von Parametern der Modaloperatoren haben, schreiben wir sie meist als Index. In einer einfachen Sortenlogik kann man diesen Index als Sorte interpretieren. Für die funktionale Übersetzung nehmen wir der Einfachheit halber an, daß die Zeit unendlich ist, d.h. die Erreichbarkeitsrelationen sind *serial*, es gibt keine Anfangs- oder Endpunkte.

Die erste Eigenschaft, die umittelbar offensichtlich ist, ist die Transitivität der Erreichbarkeitsrelationen. „Immer in der Zukunft“ bedeutet ja gerade „bis in alle Ewigkeit“, und das ist nur gewährleistet, wenn R_F transitiv ist. Das gleiche gilt für die Vergangenheitsrelation R_P und die „überhaupt immer“-Relation R_A.

Wie in den Beispielen 8 und 24 (wähle dort $p = q = F$) ausgerechnet wurde, bestehen die folgenden Korrelationen:

Lemma 33 **(Transitivität der Zeitrelationen)**

Axiomenschema: $[F]\Phi \Rightarrow [F]\,[F]\Phi$

relational: $\forall\ x,y,z\ R_F(x,y) \wedge R_F(y,z) \Rightarrow R_F(x,z)$

funktional: $\forall\ x,y_F,z_F\ \ \exists v_F\ v(x) = (y \circ z)(x)$

Das gleiche gilt für die [P] und [A]-Operatoren sowie die dazugehörigen Erreichbarkeitsrelationen und -funktionen. ■

Man geht also so vor, daß man sich erst Eigenschaften der Zeitstruktur (z.B. Transitivität) überlegt, dann ein entsprechendes modallogisches Axiomenschema *rät* [12](in diesem Fall $[F]\,\Phi \Rightarrow [F]\,[F]\Phi$), und danach mit dem SCAN-Algorithmus überprüft, ob man richtig geraten hat. Leider gibt es noch keinen Algorithmus, mit dem man sich das Erraten des Schemas sparen kann. Die Richtung Semantik $\rightarrow$ Syntax wird in diesem Fall offensichtlich deshalb bevorzugt, weil man über die Zeitstruktur eine recht gute Intuition hat. In anderen Fällen, beispielsweise bei der Definition eines Want-Operators, hat man wenig Intuition über dessen mathematische Semantik. Da ist die umgekehrte Vorgehensweise angebracht. Man überlegt sich passende Axiomenschemata und rechnet dann deren Semantik aus.

Bisher haben wir noch keinerlei Korrelation der durch die drei $\square$-Operatoren gegebenen Fragmente der Logik untereinander. Das Zukunftsfragment ist auf einer „früher-später“-Relation R_F aufgebaut und das Vergangenheitsfragment auf einer dazu bisher unabhängigen „später-früher“-Relation R_P. Daneben steht noch die R_A-Relation, die ebenfalls bisher mit den beiden anderen nichts zu tun hat. Wir versuchen also als erstes (unserer Intuition entsprechend) zu erreichen, daß die „früher-später“-Relation und die „später-früher“-Relation invers zueinander sind.

Dazu verwenden wir den einsichtigen Zusammenhang, daß man von jedem Zeitpunkt *t* mit der

---

[12] Eine Art inverser SCAN-Algorithmus, der das Raten des Axiomenschemas erspart wird derzeit ausgearbeitet.

R_F-Relation zu einem beliebigen Zeitpunkt *t'* in der Zukunft, und offensichtlich von da aus wieder in die Vergangenheit zurück zu *t* gehen kann. Als Formel geschrieben ergibt das: $\Phi \Rightarrow [F]\langle P\rangle\, \Phi$, d.h. wenn Φ *jetzt* gilt, dann gibt es für alle zukünftigen Zeitpunkte einen Zeitpunkt in der Vergangenheit, nämlich *jetzt*, wo eben Φ gilt. Wendet man auf dieses Formelschema den SCAN-Algorithmus an (eine entsprechende Erweiterung von Beispiel 14), dann ergibt sich: $\forall x,y\ R_F(x,y) \Rightarrow R_P(y,x)$. Analog erhält man mit dem dualen Schema $\Phi \Rightarrow [P]\,\langle F\rangle\Phi$ den Zusammenhang $\forall x,y\ R_P(x,y) \Rightarrow R_F(y,x)$, so daß beide zusammen tatsächlich ergeben, daß R_F und R_P invers zueinander sind (jede ist die Umkehrung der anderen).

Um den entsprechenden Zusammenhang für die Erreichbarkeitsfunktionen zu bekommen, wenden wir den SCAN-Algorithmus mit der funktionalen Übersetzung an.

Axiomenschema:	$\Phi \Rightarrow [F]\,\langle P\rangle\Phi$
negiert:	$\Phi \wedge \langle F\rangle[P]\,\neg\,\Phi$
übersetzt:	$\exists a\ P(a) \wedge \exists b_F\ \forall\ u_P\ \neg P((b \circ u)(a))$
Klauselform:	$P(a)$
	$\neg\ P((b \circ u)(a))$
P wegresolviert:	$a \neq (b \circ u)(a)$
entskolemisiert:	$\exists a,b_F\ \forall u_P\ a \neq (b \circ u)(a)$
negiert:	$\forall\ x,y_F\ \ \exists z_P\ x = (y \circ z)(x)$ ■

Analog erhält man aus dem Schema $\Phi \Rightarrow [P]\langle F\rangle\Phi$: $\forall x,y_P\ \exists z_F\ x = (y \circ z)(x)$. Das heißt, wenn man mit einer „Zukunftsfunktion“ in die Zukunft geht, dann kommt man mit einer passenden „Vergangenheitsfunktion“ wieder an den Ausgangspunkt, und umgekehrt.

Lemma 34 (Korrespondenzen zwischen [F] und [P])

Axiomenschemata:	$\Phi \Rightarrow [F]\,\langle P\rangle\ \Phi$
	$\Phi \Rightarrow [P]\,\langle F\rangle\ \Phi$
relational:	$\forall\ x,y,z\ R_F(x,y) \Leftrightarrow R_P(y,x)$
funktional:	$\forall\ x,y_F\ \ \exists z_P\ x = (y \circ z)(x)$
	$\forall\ x,y_P\ \ \exists z_F\ x = (y \circ z)(x)$ ■

Die Logik ohne den [A]-Operator könnte man als eine Art *minimale Temporallogik* betrachten. Sie ist etwas stärker als die üblicherweise in der Literatur vorkommende *minimal Tense-Logic* K_t, die erstmals von E.J. Lemmon in [Lem77] vorgestellt wurde. K_t ist ein Subsystem der von uns betrachteten Logik, welches auf die beiden Schemata der Form $\Box\Phi\Rightarrow\Box\Box\Phi$ verzichtet, also weder annimmt, daß die Zukunft der Zukunft selbst Zukunft ist, noch, daß die Vergangenheit der Vergangenheit selbst wieder Vergangenheit ist.

Die Anbindung des [A]-Operators an die beiden Zukunfts- und Vergangenheitsoperatoren ist problemlos, wenn man ausnutzt, daß etwas *immer* gilt, wenn es immer in der Zukunft gelten wird und immer in der Vergangenheit gegolten hat. Das entsprechende Schema ist:

$$[A]\ \Phi \Leftrightarrow ([F]\ \Phi \wedge [P]\ \Phi).$$

Die korrespondierenden semantischen Eigenschaften werden wieder mit dem SCAN-

Algorithmus berechnet. Man erhält folgende Korrespondenzen:

Lemma 35 (Anbindung des [A]-Operators)

Axiomenschema: $[A]\,\Phi \Leftrightarrow ([F]\,\Phi \wedge [P]\,\Phi)$

relational: $\forall x,y\ R_A(x,y) \Leftrightarrow (R_F(x,y) \vee R_P(x,y))$

funktional: $\forall x\ (\forall y_A\ \ \exists u_F, v_P\ y(x) = u(x) \vee y(x) = v(x) \wedge$
$(\forall u_F\ \ \exists y_A\ y(x) = u(x)) \wedge$
$(\forall v_P\ \ \exists y_A\ y(x) = v(x))$ ■

Wenn also z.B. die R_F-Relation als Basis gewählt wird, folgt aus den obigen Betrachtungen, daß die R_P-Relation deren Inverses ist, und weiterhin, daß die R_A-Relation die Vereinigung von R_F und R_P oder mit anderen Worten, die symmetrische Hülle von R_F ist. Dies wiederum legt nahe, nach Operatoren Ausschau zu halten, die den reflexiven Hüllen der drei Relationen entsprechen. Und in der Tat wurde bisher nicht festgelegt, ob „immer" das „jetzt" mit einschließt oder nicht. Macht man also eine Aussage $[F]\Phi$, dann ist damit nicht sicher, ob Φ auch jetzt schon gilt. Aus der Konstruktion der R_A-Relation ergibt sich noch nicht einmal, daß $[A]\Phi$ sicherstellt, daß Φ auch jetzt gilt.

Um diesem Mehrdeutigkeit abzuhelfen, könnte man entweder die vorhandenen Operatoren umdefinieren oder einfach noch neue Operatoren hinzudefinieren, so daß man beide Typen gleichzeitig verwenden kann. Wir tun das letztere und führen also drei weitere □-Operatoren mit den dazu dualen ◊-Operatoren ein.

Definition 36 (Reflexive Zeitoperatoren)

$[F^*]\Phi$	Φ gilt jetzt und immer in der Zukunft.
$[P^*]\Phi$	Φ gilt jetzt und immer in der Vergangenheit.
$[A^*]\Phi$	Φ gilt immer incl. jetzt.

Dazu gehören dann noch die dualen Operatoren

$\langle F^*\rangle\Phi$	Φ gilt jetzt oder irgendwann in der Zukunft.
$\langle P^*\rangle\Phi$	Φ gilt jetzt oder galt irgendwann in der Vergangenheit.
$\langle A^*\rangle\Phi$	Φ gilt oder galt irgendwann einmal incl. jetzt ■

Die formale Anbindung dieser Operatoren an die schon bekannten erfolgt über die Forderung, daß die den neuen Operatoren zugeordneten Erreichbarkeitsrelationen gerade die reflexive Hülle der entsprechenden Grundrelationen sind. Auf der Operatorebene entspricht dieser Forderung die Regel, daß wenn etwas immer (incl. jetzt) wahr ist, es eben immer wahr und auch jetzt wahr ist. Daraus ergeben sich unmittelbar die folgenden Korrespondenzen:

Lemma 37 (Anbindung der „jetzt und immer"-Operatoren)

Axiomenschema: $[F^*]\Phi \Leftrightarrow ([F]\Phi \wedge \Phi)$

relational: $\forall x,y\ R_{F^*}(x,y) \Leftrightarrow (R_F(x,y) \vee x = y)$

funktional: $\forall x\ (\exists z_{F^*}\ z(x) = x) \wedge$
$(\forall y_F\ \ \exists z_{F^*}\ z(x) = y(x)) \wedge$
$(\forall y_{F^*}\ \ \exists z_F\ y(x) = z(x) \vee y(x) = x)$

Die Korrelationen für die anderen beiden Operatortypen ergeben sich auf analoge Weise. ■

In der natürlichen Sprache kommen allerdings zeitliche Bezüge vor, die mit den bisher eingeführten Operatoren nicht darzustellen sind. Ein Beispiel dafür ist: *Der Krug geht solange zum Brunnen, bis er bricht*[13]. Dies liegt offensichtlich daran, daß die obigen Operatoren nur Aussagen erlauben, die entweder einzelne Zeitpunkte oder deren gesamte Zukunft (bzw. Vergangenheit) betreffen. Einzelne Zeitabschnitte lassen sich hingegen nicht abgrenzen.

Um diese Lücke zu füllen, wurden die *zweistelligen Until*- und *Since*-Operatoren **U** und **S** eingeführt. Intuitiv soll eine Formal der Form: $\Phi\ \mathbf{U}\ \Psi$ ausdrücken, daß irgendwann einmal Ψ wahr wird und in der Zwischenzeit Φ wahr ist. Analog dazu bedeutet $\Phi\ \mathbf{S}\ \Psi$, daß irgendwann mal Ψ galt und seither Φ wahr ist. Den obigen Satz könnte man dann folgendermaßen ausdrücken:

geht_zum_Brunnen(Krug) **U** bricht(Krug)

Diese informelle Semantik läßt noch offen, ob der Zeitabschnitt, in dem Φ gilt, zu einem offenen oder einem geschlossenen Intervall gehört. Darüberhinaus kann diese Interpretation noch dahingehend abgeschwächt werden, daß man nicht strikt verlangt, daß Ψ einmal gelten wird (bzw. einmal gegolten hat), sondern sich damit begnügt, daß Φ bis zu einem eventuell noch kommenden Ψ gilt und somit von jetzt ab immer gilt, falls Ψ in der Zukunft überhaupt nicht mehr auftaucht. Unter dieser Bedeutung spricht man dann von *schwachen Until*- bzw. *schwachen Since*-Operatoren. Es ist leicht einzusehen, daß sich beide Formen im allgemeinen ineinander überführen lassen.

Um die verschiedenen Versionen der Until- und Since-Operatoren syntaktisch einigermaßen übersichtlich aufschreiben zu können, führen wir folgende Notation ein:

Die Basissymbole bilden **U** und **S**. Rechts und links von diesen Symbolen werden jeweils ein Paar von Intervallklammern, mit denen man üblicherweise geschlossene, offene oder halboffene Intervalle bezeichnet, geschrieben. Für $\Phi\ l_1l_2\mathbf{U}r_1r_2\ \Psi$ bezieht sich das rechte Klammerpaar r_1r_2 auf Ψ und deutet an, daß für den Zeitpunkt z, in dem Ψ gilt, $z \in r_1\mathrm{jetzt},\infty\, r_2$ erlaubt ist, wobei $z = \infty$ bedeuten soll, daß Ψ überhaupt nicht mehr wahr werden muß (schwacher Until-Operator). Das linke Klammerpaar l_1l_2 bezieht sich entsprechend auf Φ und bedeutet, daß Φ überall in dem Intervall $l_1\mathrm{jetzt}, z l_2$ gelten muß, wobei z der Zeitpunkt ist, in dem Ψ gilt. Falls Ψ nie gilt (schwaches Until), dann muß eben Φ von jetzt an immer gelten.

Als konkretes Beispiel betrachten wir $\Phi\ []\mathbf{U}][\ \Psi$. Das bedeutet: Ψ muß bezüglich der Zukunftsrelation R_F (Gegenwart *nicht* notwendigerweise eingeschlossen) irgendwann in einem Zeitpunkt $z \neq \infty$ gelten und von jetzt an bis zu diesem Zeitpunkt z (beide Grenzen eingeschlossen) muß Φ wahr sein.

Für den Since-Operator ist die Konstruktion analog. Nur bezieht diese sich auf die Vergangenheitsrelation R_P.

---

[13]Manchen bekannt auch als Aussage über Studenten und Mensen.

Die formale Definition der Semantik von **U** und **S** ist:

Definition 38 (Semantik von Until und Since)
Zunächst führen wir folgende Notation ein:

$z \in\ ]a, \infty[_X\ := R_X(a,z)$, (X ∈ {F,P})
$z \in [a, \infty[_X\ := R_{X^*}(a,z)$, (X ∈ {F,P})
$z \in\ ]a, b[_X\ := R_X(a,z)$ und $R_X(z,b)$

Für die anderen Kombinationen gilt entsprechendes.
Für eine Interpretation $\mathfrak{I}$ mit aktuellem Zeitpunkt z sei somit:

$\mathfrak{I} \models \Phi\ l_1 l_2\mathbf{U}\ r_1[\ \Psi$ gdw. es existiert ein $x \in r_1 z, \infty[_F$ mit $\mathfrak{I}[x] \models \Psi$
und für alle x' mit $x' \in l_1 z, x\ l_{2F}$ gilt $\mathfrak{I}[x'] \models \Phi$.
$\mathfrak{I} \models \Phi\ ]l_2\mathbf{U}\ r_1]\ \Psi$ gdw. $\mathfrak{I} \models \Phi\ ]l_2\mathbf{U}\ r_1[\ \Psi$ oder $\mathfrak{I} \models [F]\ \Phi$.
$\mathfrak{I} \models \Phi\ [l_2\mathbf{U}\ r_1]\ \Psi$ gdw. $\mathfrak{I} \models \Phi\ [l_2\mathbf{U}\ r_1[\ \Psi$ oder $\mathfrak{I} \models [F^*]\ \Phi$.

Für die Since-Operatoren muß entsprechend der Index $_F$ durch $_P$ ersetzt werden. ■

Die bisherige Menge von Operatoren ist in gewisser Weise vollständig, da jeder weitere denkbare Temporaloperator durch sie beschrieben werden kann (wenigstens in linearen Zeitstrukturen). Hans Kamp [Kam68] konnte zeigen, daß dort sogar nur ein *Since*- und ein *Until*-Operator ausreichen, um beliebige zeitliche Zusammenhänge zu beschreiben. Wenn wir hier also mehr Temporaloperatoren betrachten als unbedingt nötig, so einzig und allein deswegen, weil vielen sofort ein natürlichsprachlicher Ausdruck entspricht. Da wir nicht mit den Operatoren direkt arbeiten wollen, sondern sie durch Übersetzung in Prädikatenlogik wieder loswerden, kann man sich diesen Luxus leisten.

Wir erweitern die relationale (Def. 3) und die funktionale (Def. 22) Übersetzung nun auf Until- und Since-Operatoren. Die relationale Übersetzung kann man direkt aus der Semantik ablesen. Für die funktionale ist ist dies nicht mehr ganz so offensichtlich.

Definition 39 (Übersetzung der Until- und Since-Operatoren)
Wir beginnen mit der relationalen Übersetzung.

$$\pi(\Phi\ ][\mathbf{U}][\ \Psi, z)\ =\ \exists x\ R_F(z,x) \wedge \pi(\Psi,x) \wedge \forall x'(R_F(z,x') \wedge R_F(x',x)) \Rightarrow \pi(\Phi,x')$$

Für die anderen Fälle gilt folgende Regel: Wenn sich eines den ersten drei Klammersymbole umdreht, dann ändert sich der entsprechende Index R_F zu R_{F^*}

$$\pi(\Phi\ ]l_2\mathbf{U}\ r_1]\ \Psi, z) = \pi(\Phi\ ]l_2\mathbf{U}\ r_1[\ \Psi, z) \vee \pi([F]\Phi, z).$$
$$\pi(\Phi\ [l_2\mathbf{U}\ r_1]\ \Psi, z) = \pi(\Phi\ [l_2\mathbf{U}\ r_1[\ \Psi, z) \vee \pi([F^*]\Phi, z).$$

Für die Since-Operatoren ändert sich wieder der Index $_F$ zu $_P$.

Und nun die funktionale Übersetzung:

$$\pi(\Phi\ ][\mathbf{U}][\ \Psi, z)\ =\ \exists x_F\ \pi(\Psi, x(z)) \wedge \forall u_F\ (\exists w_F\ (u \circ w)(z) = x(z)) \Rightarrow \pi(\Phi, u(z))$$

Die restlichen Fälle sind analog zur relationalen Übersetzung zu sehen. ■

2.2 Topologie der Zeit

Abgesehen von der Forderung nach der Transitivität der Zeitrelationen wurden bisher noch keine weiteren Annahmen über die Struktur der Zeit gemacht. Im folgenden untersuchen wir daher, wie man spezielle Zeitstrukturen durch Axiomenschemata bzw. Annahmen über die Zeitrelationen machen kann.

2.2.1 Linearität

Lineare und transitive Strukturen sind dadurch gekennzeichnet, daß es keine Verzweigungen gibt, bei denen von einem Punkt aus verschiedene, bezüglich der gegebenen Erreichbarkeitsrelation unvergleichbare, Punkte erreichbar sind. Diese Definition schließt kreisförmig geschlossene Strukturen oder mehrere unabhängige lineare Ketten *nicht* aus. Und in der Tat können wir mit den bisherigen Mitteln solche Modelle nicht eliminieren. Im allgemeinen stört das aber nicht, da es in den meisten Anwendungen nur darauf ankommt, echte Verzweigungen auszuschließen, um damit die Möglichkeit zu besitzen, dem $\Diamond$-Operator die Bedeutung von „unausweichlich" geben zu können (siehe Abschnitt 1.1.3.).

Es gibt verschiedene Möglichkeiten, Linearität im obigen Sinne zu erzwingen. Die erste nutzt aus, daß in linearen Strukturen zwei von der Gegenwart aus erreichbare Zeitpunkte *x* und *y* untereinander vergleichbar sind, d.h. entweder kann man von *x* aus *y*, oder umgekehrt, von *y* aus *x* erreichen. Dies genau ist der Inhalt des folgenden Axiomenschemas, welches wir wieder mit dem SCAN-Algorithmus auf korrespondierende Eigenschaften der Erreichbarkeitsrelationen und -funktionen untersuchen.

Lemma 40 (Rechtslinearität)
Axiomenschema: $\langle F\rangle\Phi \wedge \langle F\rangle\Psi \Rightarrow \langle F\rangle(\Phi \wedge \langle F\rangle\Psi) \vee \langle F\rangle(\Psi \wedge \langle F\rangle\Phi) \vee \langle F\rangle(\Phi \wedge \Psi)$
relational: $\forall x,y,z\ R_F(x,y) \wedge R_F(x,z) \Rightarrow (R_F(y,z) \vee R_F(z,y) \vee y = z)$
funktional: $\forall x,y_F,z_F\ \exists u_F\ (y \circ u)(x) = z(x) \vee (z \circ u)(x) = y(x) \vee y(x) = z(x)$ ■

Diese Axiomenschema besagt also, daß es keine echten Verzweigungen in der Zukunft geben kann. Das analoge Schema mit dem $\langle P\rangle$-Operator erzwingt dann Linearität in die Vergangenheit (Linkslinearität).

Eine Alternative bietet die Verwendung der Zukunfts- und Vergangenheitsoperatoren zusammen. Die Idee hierbei ist, daß es keine Verzweigung in die Zukunft geben kann, wenn man erst ein Stück in die Vergangenheit geht, von da wieder ein Stück in die Zukunft, und dann entweder in der Vergangenheit, Gegenwart oder Zukunft landet, aber nicht in irgendwelchen parallelen Zeitsträngen.

Lemma 41 (Rechtslinearität)
Axiomenschema: $\langle P\rangle\,\langle F\rangle\Phi \Rightarrow (\langle P\rangle\Phi \vee \Phi \vee \langle F\rangle\Phi)$
relational: $\forall\ x,y,z\ R_P(x,y) \wedge R_F(y,z) \Rightarrow (R_P(x,z) \vee R_F(x,z) \vee x = z)$
funktional: $\forall\ x,u_P,v_F\ \exists y_P,z_F\ (u \circ v)(x) = y(x) \vee (u \circ v)(x) = x \vee (u \circ v)(x) = z(x)$

Das duale Schema $\langle F\rangle\langle P\rangle\Phi \Rightarrow (\langle F\rangle\Phi \vee \Phi \vee \langle P\rangle\Phi)$ erzwingt dann konsequenterweise die

Linkslinearität. ■

Es sei an dieser Stelle noch einmal angemerkt, daß in verzweigenden Strukturen ein „unausweichlich“-Operator und in linearen Strukturen ein „möglicherweise“-Operator nicht ohne weiteres definiert werden kann. In [CE81,Ohl89] wurde versucht, wenigstens verzweigende Strukturen dahingehend zu erweitern. An dieser Stelle können wir jedoch nicht auf diese Version eingehen, da sie sich zu sehr von denen in diesem Kapitel vorgestellten Semantikdefinitionen unterscheidet.

2.2.2 Dichte Zeit

Zeitstrukturen, bei denen die Zeitpunkte ähnlich dicht liegen wie bei rationalen Zahlen auf einem Zahlenstrahl, sind sehr leicht zu erzwingen. Man verlangt einfach, daß zwischen zwei beliebigen Zeitpunkten immer mindestens ein dritter liegt. Das geht (für den Zukunftsanteil) mit dem Schema $\langle F\rangle\Phi \Rightarrow \langle F\rangle\langle F\rangle\Phi$ (vgl. Beispiel 25)

Lemma 42 (Dichte Zeitstruktur)
Axiomenschema: $\langle F\rangle\Phi \Rightarrow \langle F\rangle\langle F\rangle\Phi$
relational: $\forall x,y\ R_F(x,y) \Rightarrow (\exists z\ R_F(x,z) \wedge R_F(z,y))$
funktional: $\forall x,y_F\ \ \exists u_F,v_F\ (u \circ v)(x) = y(x)$ ■

2.2.3 Diskrete Zeit

Etwas komplizierter ist es, wenn man an diskreter Zeit interessiert ist. Man möchte dann darstellen können, daß zwischen zwei beliebigen Zeitpunkten nur endlich viele andere liegen können. Anders ausgedrückt, bedeutet dies, daß nicht unendlich viele Schritte gemacht werden können, um einen endlich weit entfernten Zeitpunkt zu erreichen. Oder, äquivalent dazu: macht man unendlich viele Schritte, ist man auch unendlich weit in die Zukunft vorgedrungen. Ein Axiomschema, welches diese Aussage repräsentiert, ist:

$$(\langle F\rangle\Phi \wedge [F](\Phi \Rightarrow \langle F\rangle\Phi)) \Rightarrow [F]\langle F\rangle\Phi$$

Zur Erklärung dieses Schemas betrachten wir als einfaches Modell die ganzen Zahlen, um die Werte 1-1/n erweitert (n sei dabei eine beliebige natürliche Zahl) und nehmen wir an, daß eine gegebene Formel Φ genau in den Welten wahr ist, die echt zwischen 0 und 1 liegen. Befindet sich das „Jetzt“ im Zeitpunkt 0, so ist die linke Seite der obigen Implikation erfüllt, die rechte Seite allerdings nicht. Wir haben hier also einen typischen Fall konstruiert, in dem auf einem endlichen Bereich unendlich oft Φ wahr sein kann und man kann sich leicht vorstellen, daß für jede Struktur in der es zwei Welten gibt zwischen denen unendlich viele weitere Welten liegen, ein solches Gegenmodell konstruiert werden kann. Das obige Schema kann also nur gültig sein in Strukturen die nirgens unendlich viele Welten zwischen zwei beliebigen Welten erlauben, oder anders ausgedrückt, in Strukturen in denen zwischen zwei beliebigen Welten nur endlich viele andere vorkommen.

Das Mittelstück des obigen Schemas $[F](\Phi \Rightarrow \langle F\rangle\Phi)$ enthält eine Rekursion. Dies deutet darauf hin, daß die korrespondierende Eigenschaft der Zeitrelationen nicht in PL1 axiomatisierbar ist. Strapaziert man den SCAN-Algorithmus in diesem Fall etwas und läßt auch höhere Ordnung zu, dann kommt man mit einiger Mühe zu folgenden Korrespondenzen.

Lemma 43 (Diskrete Zeit)
Axiomenschema: $(\langle F\rangle\Phi \wedge [F](\Phi \Rightarrow \langle F\rangle\Phi)) \Rightarrow [F]\langle F\rangle\Phi$

relational: $\forall x,y\ \forall f\ R_F(x,y) \wedge \bigwedge_{n=0}^{\infty} \left(\bigwedge_{i=0}^{n} R_F(x,f^i(y)) \Rightarrow R_F(f^n(y),f^{n+1}(y))\right)$

$\Rightarrow \forall z\ R_F(z,y) \vee\ \exists n \left(\bigwedge_{i=0}^{n} R_F(x,f^i(b)) \wedge R_F(z,f^n(y))\right)$

funktional: $\forall\ x,y_F,z_F\ \forall f\ \ \exists v\ y(x) = (z \circ v)(x) \vee (y \circ f(y(x)))(x) = (z \circ v)(x) \vee$
$(y \circ f(y(x)) \circ f(y \circ f(y(x))))(x) = (z \circ v)(x) \vee \ldots$

Die Funktion *f*, die ursprünglich von dem mittleren $\langle F\rangle$-Operator stammt, und über die quantifiziert wird, repräsentiert im Prinzip die Verzweigung der Zeitstruktur in die Zukunft. Beide Versionen, die relationale sowie die funktionale, besagen, daß man jedes *y* mit endlich vielen *f*-Schritten überholen kann. Das heißt letztendlich, daß zwischen zwei Zeitpunken nur endlich viele weitere liegen, *und daß es keine Häufungspunkte gibt.* ■

Dieses Schema drückt daher die sogenannte *starke Diskretheit* aus. Im Gegensatz dazu spricht man von *schwacher Diskretheit*, wenn man nur verlangt, daß jeder Zeitpunkt einen eindeutigen nächsten Zeitpunkt besitzt. Daß die starke die schwache Diskretheit impliziert, ist sicherlich sofort einsichtig. Das eben schon erwähnte Modell der um die Werte 1-1/n erweiterten ganzen Zahlen ist tatsächlich schwach diskret, da jede Welt einen eindeutigen Nachfolger besitzt (wenn auch keinen eindeutigen Vorgänger). Somit ist gezeigt, daß die schwache Diskretheit echt schwächer ist als die starke. Es reicht auch nicht, zu garantieren, daß jede Welt genau einen Nachfolger und genau einen Vorgänger besitzt. Ein Gegenbeispiel ist leicht dadurch konstruiert, indem wir wiederum die um 1-1/n erweiterten ganzen Zahlen betrachten, zusätzlich aber auch die Werte 1+1/n hinzufügen und die 1 dafür weglassen. Tatsächlich hat dann jede Welt einen eindeutigen Vorgänger und einen eindeutigen Nachfolger und trotzdem ist die Struktur nicht diskret.

Da in (stark und schwach) diskreten Strukturen von einem eindeutigen Nachfolger gesprochen werden kann, macht es auch Sinn, dafür einen neuen Operator einzuführen, den „Next“-Operator, im allgemeinen durch das Symbol o dargestellt. Die Eigenschaften dieses neuen Operators müssen dann natürlich auch axiomatisiert werden. Nicht sehr verwunderlich ist dabei, daß die wesentliche Eigenschaft dieses Operators sehr eng mit dem Diskretheitsaxiom von oben verwandt ist. Es ist auszudrücken, daß der Next-Operator ein Konstruktorsymbol ist, d.h. daß alleine durch sukzessive Anwendung des o alle Zeitpunkte in der Zukunft des *jetzt* erreicht werden können. Dies geschieht durch das folgende Schema:

$$\mathrm{o}\Phi \wedge [F](\Phi \Rightarrow \mathrm{o}\Phi) \Rightarrow\ [F]\Phi$$

übersetzt man das Schema funktional, indem man den initialen Zeitpunkt mit 0 bezeichnet und dem o-Operator die Funktion +1 zuordnet, dann ergibt sich mit

$$P(1) \wedge \forall x\, (P(x) \Rightarrow P(x+1)) \Rightarrow \forall x\, P(x)$$

das bekannte Induktionsaxiom. Damit ist man also endgültig in Logik 2. Stufe.

2.2.4 Endliche oder unendliche Zeit

Alle bisher betrachteten Axiome machten keinerlei Aussagen darüber, ob Zeit endlich oder unendlich ist. Dies bedeutet insbesondere, daß alles, was aus den obigen Axiomen folgt, sowohl in endenden als auch in nicht endenden Strukturen gilt. Es mag allerdings vorkommen, daß man nur an unendlichen Zeitgeraden interessiert ist. Dies läßt sich nun sehr einfach durch das Serialitätsschema $[\mathsf{F}]\Phi \Rightarrow \langle \mathsf{F} \rangle \Phi$ axiomatisieren, welches nichts weiter ausdrückt als: die Menge der Nachfolger eines beliebigen Zeitpunktes ist nicht leer.

Möchte man aber im Gegensatz dazu die Endlichkeit der Zeit erzwingen, kann man stattdessen das Schema: $\langle \mathsf{F}^* \rangle [\mathsf{F}]\mathsf{falsch}$ hinzufügen. Mit dessen Hilfe wird ausgesagt, daß es einen Zeitpunkt gibt, von dem ab falsch wahr ist. Dies ist nur dann nicht paradox, wenn dieser ausgezeichnete Zeitpunkt gar keine Nachfolger mehr besitzt, die Zeit also in diesem Moment endet. Das dazugehörige Korrespondenzaxiom ist $\forall x\, \exists y\, R_F(x,y) \wedge \forall z\, \neg R_F(y,z)$, was genau besagt, daß es für jeden Zeitpunkt x einen letzten Zeitpunkt y gibt.

Auf die gleiche Weise läßt sich natürlich auch der Beginn der Zeit darstellen, und zwar indem man gerade das Spiegelbild $\langle \mathsf{P}^* \rangle [\mathsf{P}]\mathsf{falsch}$ zur Axiomenmenge hinzufügt.

Neben den hier betrachteten Eigenschaften werden in der Literatur hin und wieder auch andere Eigenschaften erwähnt, wie zum Beispiel die *Zirkularität* oder die *Kontinuität*. Genaueres findet man in [RU71, Pri67].

2.3 METATEM

Es gibt etliche Ansätze, Temporallogik als Programmiersprache zu verwenden [Brz89, Bau89]. Stellvertretend dafür werden wir kurz die Idee des Systems METATEM beschreiben. METATEM beruht auf USF-Logik, einer von Dov Gabbay entwickelten Version von linearer diskreter Temporallogik [BG91,Gab89]. Verschiedene Versionen des METATEM-Systems werden derzeit an verschiedenen Instituten entwickelt [BFGGO89].

Ein METATEM-Programm besteht im wesentlichen aus Formeln der Art

$$\Box(\mathsf{Past} \Rightarrow \mathsf{Future}),$$

wobei Past eine Vergangenheitsformel und Future eine Zukunftsformel (einschließlich Gegenwart) ist[14].

Die Datenbasis, über der Programme dieser Art arbeiten, besteht aus einem partiellen Modell.

[14] Vergangenheitsformeln enthalten keine Zukunftsoperatoren und Zukunftsformeln enthalten keine Vergangenheitsoperatoren.

Das heißt, für eine Sequenz von diskreten Zeitpunkten von irgendeinem Punkt in der Vergangenheit aus bis jetzt wird genau festgehalten, welches Faktum wahr und welches falsch ist. Über diesem Modell können jetzt die Vorbedingungen der METATEM-Regeln, d.h. die Vergangenheitsteile, ausgewertet werden. Für alle Regeln, deren Vorbedingungen erfüllt sind, muß jetzt versucht werden, die rechte Seite, d.h. die Gegenwarts- und Zukunftsanteile *wahr zu machen*. Das heißt, es wird ausgerechnet, wie die Atome konfliktfrei wahr gemacht werden können. Wahr machen heißt dabei, einmal das Literal in die Datenbasis als wahr einzutragen, d.h. das partielle Modell weiter in die Zukunft auszudehnen, und zum anderen eine dem Literal zugeordnete externe Prozedur aufzurufen, die irgendwelche beliebige Aktionen durchführt. Wenn z.B. eine Regel sagt, $\langle F^* \rangle P$ und eine andere sagt $\langle F^* \rangle \neg P$, dann läßt sich das konfliktfrei wahr machen, indem man z.B. nach dem Motto „was du heute kannst besorgen, das verschiebe nicht auf morgen“ P sofort wahr macht — ausführt — und $\neg P$ erst beim nächstmöglichen Zeittick.

Der Interpretierer von METATEM verwendet dabei spezielle Regeln, die nur in diskreten Strukturen gelten. Zwei davon sind:

$$\langle F^* \rangle\, \Phi \Leftrightarrow (\Phi \vee \mathrm{o}\langle F^* \rangle\, \Phi)$$
$$[F^*]\, \Phi \Leftrightarrow (\Phi \wedge \mathrm{o}[F^*]\, \Phi)$$

D.h. wenn etwas irgendwann gelten soll, dann kann es entweder gleich gelten oder vom nächsten Zeitpunkt aus irgendwann gelten, und wenn etwas immer gelten soll, dann muß es jetzt gleich gelten und darüberhinaus vom nächsten Zeitpunkt aus immer gelten. Mit solchen Regeln ist es möglich, jede Zukunftsformel aufzuspalten in einen Teil, der angibt, was sofort zu tun ist, und einen Teil, der angibt, was auf später verschoben werden kann.

Das „Herz“ des METATEM-Systems besteht aus einer Funktion **execute**, welche als Eingabe einen gegenwärtigen Zustand, eine Regelmenge, das bis dahin konstruierte Modell sowie die aus den vorhergegangen Schritten noch wahr zu machenden Formeln erhält und einen neuen Zustand liefert. Dabei führt sie im wesentlichen die folgenden Schritte durch:

1. Sammle die rechten Seiten der Regeln, deren linke Seiten durch das bisherige Teilmodell erfüllt sind.
2. Transformiere diese Formeln mit den übertragenen Bedingungen in Formeln der Form
$$\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m} (p_{ij} \wedge \neg q_{ij}) \wedge \mathrm{o}\Phi_i,$$
wobei der Teil $\mathrm{o}\Phi_i$ auch leer sein kann und alle p_{ij} und q_{ij} Literale sind.
3. Wähle eines dieser disjunktiven Elemente aus.
4. Konstruiere daraus einen neuen aktuellen Zustand der Datenbasis und bestimme die in den nächsten Zyklus zu übertragenden Bedingungen.
5. Beginne wieder beim ersten Schritt.

Man kann diese Abarbeitung also als eine Art Produktionsregelsystem sehen, in dem die diversen Bedingungsteile temporallogische Ausdrücke sind. Der „Meta“-Aspekt dabei ist, daß sich die Funktion **execute** wieder in METATEM selbst schreiben läßt, und damit eine universelle Programmiersprache zur Verfügung steht. Dieser Aspekt ist allerdings noch nicht vollständig realisiert.

Als konkrete Anwendung wurde bisher das PAYE-System (Pay As You Earn) entwickelt, in welchem das britische Personalfinanz- und Steuerrecht kodiert ist [TM90]

3 Zusammenfassung

Wir haben in diesem Kapitel gezeigt, was Modallogiken sind, was man damit modellieren kann und wie man sie manipuliert, um sie für spezielle Anwendungen anzupassen. Mit parametrisierten Modaloperatoren ergibt sich die Möglichkeit, mehrere Interpretationen (Zeit, Aktion, epistemische Interpretation usw.) in eine Logik zu integrieren. Der SCAN-Algorithmus gibt dann die Möglichkeit, die als Axiomenschemata formulierten Korrelationen zwischen den verschiedenen Parametern zu untersuchen und, falls möglich, in prädikatenlogische Axiome zu übersetzen.

Mit Hilfe der Semantik der Modaloperatoren lassen sich Modalformeln in Prädikatenlogik übersetzen und dann mit prädikatenlogischen Inferenzmethoden weiter bearbeiten. Weiterhin wurde gezeigt, wie man eine Semantikdefinition für die Modaloperatoren so umbaut, daß die Übersetzung in Prädikatenlogik kompaktere Formeln produziert, die dann effizienter zu verarbeiten sind.

Weiterhin haben wir erläutert, wie man die Semantik so abschwächen kann, daß eingebaute Gesetze wie die Notwendigkeitsregel ihre Gültigkeit verlieren. Dadurch ergeben sich mehr Möglichkeiten, durch Ein- und Ausschalten von Axiomenschemata die Logik für spezielle Anwendungen anzupassen.

Wie man mit diesen Ideen und Mechanismen neue Logiken konstruiert, wurde ausführlich am Beispiel Temporallogik behandelt.

Wir hoffen, damit die wesentlichen Ideen zur Logik selbst und zu ihren Manipulationsmöglichkeiten vermittelt zu haben. Der Leser sollte nun in der Lage sein, selbst neue Varianten zu definieren und zu untersuchen.

Literatur

Bau89 M. Baudinet. *Temporal logic programming is complete and expressive*. In Proc. of ACM Symposium on Principles of Programming Languages. University of Austin, Texas, 1989.

BFGGO89 H. Barringer, M. Fischer, D.M. Gabbay, G. Gough, R. Owens. *METATEM a framework for programming in logic*. In REX Workshop on Stepwise Refinement of Distributed Systems: Models Formalisms, Correctness (LNCS Vol. 430), S. 94-129. Springer Verlag, 1989.

BG91 H. Barringer, D.M. Gabbay. *The imerative future: An executable view of temporal logic*. The third european summer school in language, logic and information, Universität Saarbrücken, 1991.

Brz89 C. Brzoska. Temporal logic ptogramming, a survey. Interner Bericht 22/89, Fakultät für Informatik, University of Karlsruhe, 1989.

CE81 M.C. Clarke, E.A. Emerson. *Design and synthesis of synchronization skeletons using branching time temporal logic*. In Lecture Notes in Computer Science 131, S. 52-71. Springer Verlag, 1981.

Che80 B. F. Chellas. *Modal Logic: An Introduction*. Cambridge University Press, Cambridge, 1980.

Fit83 M. C. Fitting. *Proof Methods for Modal and Intuitionistic Logics*, volume 169 of Synthese Library. D. Reidel Publishing Company, 1983.

Gab89 D.M. Gabbay. *Declarative past and imperative future: Executable temporal logic for interactive systems*. In B. Banieqbal, H. Barringer, and A. Pnueil, editors, Proc. of Colloquium on Temporal Logic in Specification (LNCS, Vol. 398), S. 402-450. Springer Verlag, 1989.

GO92 D.M. Gabbay, H.J. Ohlbach. *Automated synthesis of correspondence axioms*. Max Planck Institute für Informatik, Saarbrücken, 1992. To be published.

HC68 G.E. Hughes, M.J. Cresswell. *An Introduction to Modal Logic*. Routledge, London, New York, 1968.

Hen61 L. Henkin. *Some remarks on infinitely long formulas*. In Infinistic Methods, S. 167-183. Pergamon Press, Oxford, 1961.

Her89 A. Herzig. *Raisonnement automatique en logique modale et algorithmes d'unification.* PhD thesis, Université Paul-Sabatier, Toulouse, 1989.

Kam68 H. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, 1968.

Kri59 S. A. Kripke. *A completeness theorem in modal logic*. Journal of Symbolic Logic, 24:1-14, 1959.

Kri63 S. A. Kripke. *Semantical analysis of modal logic I, normal propositional calculi.* Zeitschrift für mathematische Logik und Grundlagen der Mathematik, 9:67-96, 1963.

Lem77 E.J. Lemmon. *An Introduction to Modal Logic*, volume 11 of American Philosophical Quarterly Monograph Series. Basil Blackwell, Oxford, 1977.

Moo80 R.C. Moore. *Reasoning about Knowledge and Action*. PhD thesis, MIT, Cambdridge, 1980.

Ohl88 H.J. Ohlbach. *A resolution calculus for modal logics*. In Ewing Lusk and Ross Overbeek, editors, Proc. of 9th International Conference on Automated Deduction, CADE-88 Argonne, IL, volume 310 of Lecture Notes in Computer Science, S. 500-516, Berlin, Heidelberg, New York, 1988. Springer-Verlag. extended version: SEKI Report SR-88-08, FB Informatik, Universität Kaiserslautern, 1988.

Ohl89 H.J. Ohlbach. *Context Logic*. SEKI Report SR-89-08, FB Informatik, Universität Kaiserslautern, Germany, 1989.

Ohl91 H.J. Ohlbach. *Semantics based translation methods for modal logics*. Journal of Logic and Computation, 1(6), 1991.

Pri67 A. Prior. *Past, Present and Future*. Clarendon Press, Oxford, 1967.

Rau79 W. Rautenberg. *Klassische und nichtklassische Aussagenlogik*. Vieweg Verlag, Braunschweig, 1979.

RU71 N. Rescher, A. Urquhart. *Temporal Logic*. Springer Verlag, 1971.

TM90 I. S. Torsun, K. J. Manning. *Execution and application of temporal modal logic*. Internal report, University of Bradford, Dept. of Computing, 1990.

Ben83 J. v. Benthem. *The Logic of Time*. Reidel, Kluwer Academic Publishers, Dordrecht, 1983.

Ben84 J. v. Benthem. *Correspondence theory*. In Gabbay Dov M and Franz Guenthner, editors, Handbook of Philosophical Logic, Vol. II, Extensions of Classical Logic, Synthese Library Vo. 165, S. 167-248. D. Reidel Publishing Company, Dordrecht, 1984.

Wes89 D. Westerståhl. *Quantifiers in formal and natural language*. In Dov M. Gabbay and Franz Guenthner, editors, Handbook of Philosophical Logic, Volume IV, Topics in the Philosophy of Language, Synthese Library Vol. 167, S. 1-132. Kluwer Academic Publishers, Dordrecht, 1989.