Nichtklassische Logiken

P. H. Schmitt

Juli 2004

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Wiederholung

1.1 Wiederholung des Prädikatenkalküls

Eine Erklärung des Prädikatenkalküls erster Stufe umfaßt drei Teile:

- einen syntaktischen Teil; welche Symbole werden zum Aufbau der Sprache benutzt, welche Kombinationen dieser Symbole sind in der Sprache des Prädikatenkalküls erster Stufe erlaubt?
- einen semantischen Teil; worüber kann man in der Sprache des Prädikatenkalküls reden, welche Strukturen kann man mit seiner Hilfe untersuchen?
- einen Teil, der den Zusammenhang herstellt zwischen Syntax und Semantik in Form einer Wahrheitsdefinition; wann ist eine Formel des Prädikatenkalküls in einer Struktur wahr?

Die Syntax des Prädikatenkalküls erster Stufe

Zunächst muß darauf hingewiesen werden, daß es die Sprache des Prädikatenkalküls nicht gibt, sondern eine ganze Familie von solchen Sprachen in Abhängigkeit von einem Vokabular V.

Definition 1 *Vokabular*

- *einer Menge von Funktionszeichen Fkt(V)*
- *einer Menge von Konstantenzeichen Kon(V)*
- *einer Menge von Prädikatszeichen Pr(V)*

Das Vokabular V darf die leere Menge $\emptyset$ sein.

Neben dem Vokabular gibt es noch einen zweiten Teil von Symbolen, die sogenannten logischen Symbole, die in jeder Sprache des Prädikatenkalküls vorhanden sind:

- die aussagenlogischen Junktoren $\wedge$, $\vee$, $\leftarrow$, $\neg$
- die Quantoren $\forall x$, $\exists x$
- die Variablen x, y, z, x_i, y_j etc.

- Klammern (,)

In der Regel wird auch das Gleichheitszeichen „=“ zu den logischen Symbolen gezählt. Wir wollen das aus zwei Gründen nicht tun: Erstens ist die Konstruktion von Herbrand-Universen im Prädikatenkalkül ohne Gleichheit etwas einfacher, und zweitens ist es ein leichtes, das Gleichheitszeichen zum Vokabular V hinzuzufügen und durch die Wahl geeigneter Gleichheitsaxiome sicherzustellen, daß dieses Zeichen fast wie die wirkliche Gleichheit interpretiert wird.

Die nächste syntaktische Einheit im Prädikatenkalkül bilden die Funktionsterme, die wie folgt definiert sind:

Definition 2 (Funktionsterme)

1. *Konstanten sind Funktionsterme.*
2. *Variable sind Funktionsterme.*
3. *sind $t_1, \ldots, t_k$ Funktionsterme und f ein k-stelliges Funktionszeichen, dann ist $f(t_1, \ldots, t_k)$ wieder ein Funktionsterm.*

Mit Hilfe der Funktionsterme lassen sich atomare Formeln bilden. Atomare Formeln sind genau die Zeichenfolgen der Form:

$$p(t_1, \ldots, t_n)$$

wobei p ein n-stelliges Prädikatszeichen ist und $t_1, \ldots, t_n$ Funktionsterme sind. Die Menge der atomaren Formeln wird mit $At(V)$ bezeichnet.

Die Menge der Formeln des Prädikatenkalküls über dem Vokabular V, bezeichnet mit Fml(V), schließlich ist wie folgt rekursiv definiert:

Definition 3 *Formeln*

1. *Jede atomare Formel ist eine Formel.*
2. *Sind A, B Formeln, so auch*
   - $(A \wedge B)$ *gelesen: A und B*

- $(A \vee B)$ *gelesen: A oder B*
- $(A \leftarrow B)$ *gelesen: wenn B, dann A*
- $\neg A$ *gelesen: nicht A*

3. $(\exists x A)$ *gelesen: es gibt x, so daß A*

4. $(\forall x A)$ *gelesen: für alle x gilt A*

Es hat sich in der Literatur zur logischen Programmierung die Konvention eingebürgert, den Implikationspfeil von rechts nach links zu orientieren. Die Aussage „wenn A, dann B“ wird also durch $B \leftarrow A$ formalisiert und nicht wie traditionell in der Logik gebräuchlich durch $A \rightarrow B$. Das ist jedoch ein rein notationeller Unterschied, so wie wir die Tatsache, daß 2 kleiner als 5 ist, sowohl in der Form $2 < 5$ als auch in der Form $5 > 2$ notieren können.

Der Doppelpfeil $\Leftrightarrow$ gehört nicht zu den logischen Symbolen. Wir benutzen gelegentlich $A \leftrightarrow B$ als abkürzende Schreibweise für $(A \leftarrow B) \wedge (B \leftarrow A)$.

Substitution

Eine Substitution s ist eine Abbildung von der Menge aller Variablen in die Menge aller Terme über einem gegebenen Vokabular V.

Sei A eine quantorenfreie Formel. Die Formel $s(A)$ entsteht aus A durch Substitution von x_i durch $s(x_i)$. Formal wird $s(A)$ rekursiv definiert. Zunächst für Funktionsterme t:

$$s(t) = \begin{cases} s(x) & \text{falls } t \text{ eine Variable } x \text{ ist} \\ c & \text{falls } t \text{ eine Konstante } c \text{ ist} \\ f(s(t_1), \ldots, s(t_k)) & \text{falls } t = f(t_1, \ldots, t_k) \end{cases}$$

Für eine atomare Formel $A(t_1, \ldots, t_k)$ definieren wir

$$s(A) = A(s(t_1), \ldots, s(t_k)).$$

Für zusammengesetzte Formeln haben wir, was schon fast selbstverständlich ist:

- $s(A \wedge B) = s(A) \wedge s(B)$
- $s(A \vee B) = s(A) \vee s(B)$

- $s(A \to B) = s(A) \to s(B)$
- $s(\neg A) = \neg s(A)$

Man kann Substitutionen auch für Formeln mit Quantoren definieren. Dabei wird man jedoch Variablen, die durch einen Quantor gebunden werden, unterscheiden von Variablen, auf die das nicht zutrifft, den sog. freien Variablen. Wir kommen fürs erste mit Substitutionen in quantorenfreien Formeln aus.

Der Definitionsbereich einer Substitution s, bezeichnet mit $Def(s)$, ist die Menge aller Variablen, die s nicht auf sich selbst abbildet.

$Def(s) = \{x : s(x) \neq x\}$

Die Menge

$Wb(s) = \{s(x) : x \in Def(s)\}$

heißt der Wertebereich der Substitution s.

In allen Fällen, die wir betrachten, wird der Definitionsbereich von s endlich sein. Eine solche Substitution s geben wir durch die Menge der Paare

$$\{\langle x, s(x)\rangle \mid s(x) \neq x\}$$

an.

Beispiel 1 *Die Substitution s sei gegeben durch*

$$s = \{< x, f(y) >, < y, c >\}.$$

Dann ergibt sich für den Term $t = h(x, f(y))$ $s(t) = h(f(y), f(c))$.

Bemerkung: Es werden nur die Vorkommen von Variablen im Ausgangsterm bzw. in der Ausgangsformel ersetzt. Im obigen Beispiel entsteht nach der Ersetzung von x durch $f(y)$ zunächst $h(f(y), f(y))$. In der zweiten Auswertung von s wird nur im zweiten Vorkommen von $f(y)$ die Variable y durch die Konstante c ersetzt. Man spricht deshalb auch von simultaner Substitution, was suggerieren soll, daß alle Variablenersetzungen simultan geschehen, so daß die Versuchung, ein Variablenvorkommen zu substituieren, das im Ausgangsterm nicht vorhanden war, erst gar nicht auftritt.

Zur Vereinfachung der Notation schreiben wir häufig $A(t_1, \ldots, t_k)$ oder etwas detaillierter $A(t_1/x_1, \ldots, t_k/x_k)$ für $s(A)$, wobei s durch die Menge $\{\langle x_1, t_1 >, \ldots, < x_k, t_k \rangle\}$ gegeben ist.

Semantik des Prädikatenkalküls erster Stufe

Zur Angabe einer modelltheoretischen Semantik für $Fml(V)$ wählt man zunächst einen Bereich U von Elementen aus, über die man reden möchte. Jedem Konstantenzeichen c aus V ordnet man ein Element $c^{\mathcal{H}} \in U$ zu, jedem k-stelligen Funktionszeichen f aus V eine Funktion $f^{\mathcal{H}} : U^k \rightarrow U$ und jedem k-stelligen Prädikatszeichen p aus V eine Menge $p^{\mathcal{H}} \subseteq U^k$.
Das so entstehende Gebilde

$$\mathcal{H} =< U, c^{\mathcal{H}} : c \in Kon(V), f^{\mathcal{H}} : f \in Fkt(V), p^{\mathcal{H}} : p \in Pr(V) >$$

nennen wir eine Struktur für V.

Wir wollen einige Beispiele für Strukturen betrachten, um einen Eindruck von der Flexibilität dieses Konzepts zu gewinnen.

Als erstes geben wir uns ein Vokabular V_1 vor, bestehend aus:

Konstantensymbole	a, nil
Funktionszeichen	$f(\_, \_)$
Prädikatszeichen	$d(\_, \_, \_)$

Beispiel 2
Die Struktur $\mathcal{H}_1$ für V_1 besteht aus:

U_1	$=$	*die Menge aller natürlichen Zahlen $\cup$ die Menge aller endlichen Folgen natürlicher Zahlen.*
$nil^{\mathcal{H}_1}$	$=$	*die leere Folge*
$a^{\mathcal{H}_1}$	$=$	1
$f^{\mathcal{H}_1}(a, b)$	$=$	$< a, b_1, \ldots, b_k >$ *falls a eine natürliche Zahl ist und b die Folge* $< b_1, \ldots, b_k >$
$f^{\mathcal{H}_1}(a, b)$	$=$	*die leere Folge in allen anderen Fällen.*
$d^{\mathcal{H}_1}(a, b, c)$	*gdw*	*b eine natürliche Zahl ist, a, c Listen sind, c aus a durch Wegfall eines Vorkommens von b entsteht. Kommt b in der Folge a nicht vor, so muß $c = a$ sein.*

Die Struktur $\mathcal{H}_2$ für das Vokabular V_1 könnte wie folgt aussehen:

Beispiel 3

$$\begin{array}{lll}
U_2 & = & \textit{die Menge aller natürlichen Zahlen.} \\
nil^{\mathcal{H}_2} & = & 0 \\
a^{\mathcal{H}_2} & = & 1 \\
f^{\mathcal{H}_2}(a,b) & = & a * b \textit{ (das Produkt von a und b)} \\
d^{\mathcal{H}_2}(a,b,c) & gdw & b * c = a.
\end{array}$$

Schließlich sei die dritte Struktur $\mathcal{H}_3$ für V_1 wie folgt gegeben:

Beispiel 4

$$\begin{array}{lll}
U_3 & = & \textit{Menge aller Terme für } V_1. \\
nil^{\mathcal{H}_3} & = & nil \\
a^{\mathcal{H}_3} & = & a \\
f^{\mathcal{H}_3}(a,b) & = & f(a,b) \\
d^{\mathcal{H}_3}(a,b,c) & gdw & a = f(a_1, f(a_2, \ldots f(a_k, nil)) \ldots) \\
 & & c = f(c_1, f(c_2, \ldots f(c_{k-1}, nil)) \ldots) \\
 & & \{c_i : i < k\} \cup \{b\} = \{a_i : i < k\} \\
 & & \textit{falls } a = nil \textit{ ist oder } b \textit{ nicht in } a \textit{ vorkommt, so} \\
 & & \textit{muß } c = a \textit{ sein.}
\end{array}$$

Die Interpretation von Formeln des Prädikatenkalküls

Wir fixieren für die folgenden Erklärungen eine Struktur M für ein Vokabular V. Eine Variablenbelegung b ist eine Funktion von der Menge aller Variablen in das Universum der betrachteten Struktur M.

In Abhängigkeit von der Struktur M für V und einer Variablenbelegung β ordnet man in naheliegender Weise jedem Funktionsterm t, der nur Zeichen aus V enthält, ein Element $I(\mathcal{M}, \beta)(t)$ aus U zu. Die formale Definition von $I(\mathcal{M}, \beta)(t)$ erfolgt rekursiv parallel zur rekursiven Definition von Funktionstermen.

Definition 4 *Interpretation von Termen*

$$\begin{array}{lll}
I(\mathcal{M}, \beta)(t) & = & \beta(t), \textit{ falls } t \textit{ eine Variable ist} \\
I(\mathcal{M}, \beta)(t) & = & t^{\mathcal{M}}, \textit{ falls } t \textit{ eine Konstante ist} \\
I(\mathcal{M}, \beta)(f(t_1, \ldots, t_k)) & = & f^{\mathcal{M}}(I(\mathcal{M}, \beta)(t_1), \ldots, I(\mathcal{M}, \beta)(t_k))
\end{array}$$

Definition 5 *Interpretation von Formeln*

Für eine atomare Formel $A = p(t_1, \ldots, t_n)$, die nur Zeichen aus V enthält, gilt $(\mathcal{M}, \beta) \models A$ genau dann, wenn das Elementtupel ($I(\mathcal{M}, \beta)(t_1)$,...,$I(\mathcal{M}, \beta)(t_n)$) in $I(\mathcal{M})(p)$ liegt. Für zusammengesetzte Formeln gilt

$(\mathcal{M}, \beta) \models (A \wedge B)$	*gdw*	$\mathcal{M}, \beta) \models A$ *und* $(\mathcal{M}, \beta) \models B$
$(\mathcal{M}, \beta) \models (A \vee B)$	*gdw*	$(\mathcal{M}, \beta) \models A$ *oder* $(\mathcal{M}, \beta) \models B$
$(\mathcal{M}, \beta) \models (A \leftarrow B)$	*gdw*	*nicht* $(\mathcal{M}, \beta) \models B$ *oder* $(\mathcal{M}, \beta) \models A$
$(\mathcal{M}, \beta) \models \neg A$	*gdw*	*nicht* $(\mathcal{M}, \beta) \models A$
$(\mathcal{M}, \beta) \models \exists x A$	*gdw*	*es gibt ein Element u aus dem Universum U, so daß* $(\mathcal{M}, \beta') \models A$, *wobei* $\beta'(y) = \beta(y)$ *falls* $y \neq x$ $\beta'(y) = u$ *falls* $y = x$.
$(\mathcal{M}, \beta) \models \forall x A$	*gdw*	*für alle u aus dem Universum U von $\mathcal{M}$ gilt:* $(\mathcal{M}, \beta') \models A$ *wobei β' wie oben aus β entsteht.*

Bei der Berechnung von $I(\mathcal{M}, \beta)(t)$ und $(\mathcal{M}, \beta) \models A$ kommt es nur auf die Belegung der Variablen an, die tatsächlich in t bzw. A vorkommen. Das sind immer endlich viele. Wir schreiben deshalb meistens die endlich vielen Werte aus U, die zur Belegung dieser Variablen vorgesehen sind, direkt hin, anstatt β. Das sieht dann so aus:

$$I(\mathcal{M}, n_1, \ldots, n_k)(t)$$

$$\mathcal{M} \models A(n_1, \ldots, n_k)$$

wobei die Information, welcher der Werte $n_1, \ldots, n_k$ welcher Variablen zugeordnet werden soll, aus dem Kontext ersichtlich sein wird. Meist sind die auftretenden Variablen durch Indizes und/oder alphabetisch geordnet und n_1 wird der in diesem Sinne ersten, n_2 der zweiten Variablen zugeordnet usw.

Beispiel 5
$I(\mathcal{H}_1, 5, <2, 4, 1>)(f(a, b)) = <5, 2, 4, 1>$
$I(\mathcal{H}_2, 5, 2)(f(a, b)) = 10$
$\mathcal{H}_1 \models d(a, b, c)(<5, 7, 2, 7>, 7, <5, 2, 7>)$
Die folgende Aussage ist im Unterschied zu den vorangegangenen falsch:
$\mathcal{H}_1 \models d(a, b, c)(<5, 7, 2, 7>, 7, <5, 2>)$

Der universelle Abschluß einer Formel A wird gebildet, indem für jede Variable x in A, die nicht schon im Bereich eines Quantors steht, der Allquantor $\forall x$ der Formel A vorangestellt wird. Ist D eine Menge von Formeln und A eine einzelne Formel in $Fml(V)$, so nennen wir A eine Folgerung oder eine Konsequenz aus D, in Symbolen

$$D \vdash A$$

, wenn für jede Struktur $\mathcal{M}$ für V, so daß für den universellen Abschluß C^* jeder Formel C in D $\mathcal{M} \models C^*$ gilt, auch $\mathcal{M} \models A^*$ gilt für den universellen Abschluß A^* von A.

Gilt $\mathcal{M} \models D$, so nennen wir $\mathcal{M}$ ein Modell für D. Ist D die leere Menge, so schreiben wir $\vdash A$ anstelle von $D \vdash A$. Ein solches A heißt allgemeingültig oder eine Tautologie. Zwei Formeln A, B heißen äquivalent, wenn sowohl $A \leftarrow B$ als auch $B \leftarrow A$ allgemeingültig sind. Wir schreiben dafür $A \leftrightarrow B$. Anstelle von A ist äquivalent zu B, könnten wir auch sagen, $A \leftrightarrow B$ ist eine Tautologie.

1.2 Logik zweiter Stufe

In der Logik zweiter Stufe, die wir hier betrachten wollen, kommen gegenüber der Prädikatenlogik erster Stufe zustätzlich Variablensymbole für Mengen vor, die wir mir großen Buchstaben vom Ende des Alphabets bezeichnen, X,Y, X_i. . . .

Definition 6 *Formeln zweiter Stufe*

1. *Jede Formel der Logik erster Stufe*
2. *Ist X eine Mengenvariable, und t ein Funktionsterm, dann ist $X(t)$ eine Formel der Logik zweiter Stufe. (siehe Definition 3) ist auch eine Formel der Logik zweiter Stufe.*
3. *Ist A eine Formel der Logik zweiter Stufe, dann auch* $(\exists X A)$
4. *Ist A eine Formel der Logik zweiter Stufe,* $(\forall X A)$

Es gibt viele Varianten einer Logik zweiter Stufe. Wir beschränken uns hier auf die Quantifizierung von Mengen, was gleichbedeutend ist mit der Quantifizierung einstelliger Relationen. In anderen Versionen der Logik kann man auch mehrstellige Relationen oder auch Funktionen quantifiziert.

Eine Struktur $\mathcal{M}$ für eine Logik zweiter Stufe zum Vokabular V unterscheidet sich nicht von einer Struktur für die Prädikatenlogik erster Stufe. Zur Auswertung einer Formel A der Logik zweiter Stufe in einer Struktur $\mathcal{M}$ braucht man jetzt allerdings neben der Belegung der freien Individuenvariablen β noch eine Belegung γ der freien Mengenvariablen, die jeder Mengenvariablen X eine Teilmenge $\gamma(X) \subseteq M$ des Universums von $\mathcal{M}$ zuordnet.

Definition 7 *Interpretation von Formeln zweiter Stufe*

Sei $\mathcal{M}$ eine Struktur, β eine Belegungsfunktion der Individuenvariablen, γ eine Belegungsfunktion der Mengenvariablen. Fr eine Formel A der Logik zweiter Stufe ist $(\mathcal{M}, \beta, \gamma) \models A$ wie in Definition 5 definiert mit den zustzliche Klauseln

$$\begin{array}{lll}
(\mathcal{M}, \beta, \gamma) \models X(t) & \textit{gdw} & I(\mathcal{M}, \beta)(t) \in \gamma(X) \\
(\mathcal{M}, \beta, \gamma) \models \exists X A & \textit{gdw} & \textit{es gibt eine Teilmenge } S \textit{ des Universums } U, \\
 & & \textit{so daß } (\mathcal{M}, \beta, \gamma') \models A, \textit{ wobei} \\
 & & \gamma'(Y) = \gamma(y) \textit{ falls } Y \neq X \\
 & & \gamma'(Y) = S \textit{ falls } Y = X. \\
(\mathcal{M}, \beta, \gamma) \models \forall X A & \textit{gdw} & \textit{für alle Teilmengen } S \textit{ von } U \textit{ gilt:} \\
 & & (\mathcal{M}, \beta, \gamma') \models A \\
 & & \textit{wobei } \gamma' \textit{ wie oben aus } \gamma \textit{ entsteht.}
\end{array}$$

1.3 Grundlegende Definitionen

Definition 8 (Äquivalenzrelation) *Eine Relation R auf einer Menge M heißt eine Äquivalenzrelation, wenn sie folgende drei Eigenschaften erfüllt*

1. *für alle $m \in M$ gilt $R(m, m)$*
   (Reflexivität)

2. *für alle $m_1, m_2 \in M$ folgt aus $R(m_1, m_2)$ auch $R(m_2, m_1)$*
   (Symmetrie)

3. *für alle* $m_1, m_2, m_3 \in M$ *mit* $R(m_1, m_2)$ *und* $R(m_2, m_3)$ *gilt auch* $R(m_1, m_3)$
   (Transitivität)

Definition 9 (Kongruenzrelation) *Sei* $\mathcal{M}$ *eine prädikatenlogische Struktur zum Vokabular* V*. Eine Äquivalenzrelation* E *auf dem Universum* M *von* $\mathcal{M}$ *heißt eine Kongruenzrelation, wenn für jedes Funktionszeichen* f *der Stelligkeit* n *aus dem Vokabular und je zwei* n*-Tupel* $m_1, \ldots, m_n$*,* $m'_1, \ldots, m'_n$ *von Elementen aus* M *mit* $E(m_i, m'_i)$ *für alle* $i = 1, \ldots, n$ *gilt auch* $E(f(m_1, \ldots, m_n), f(m'_1, \ldots, m'_n))$

Definition 10 (Funktionen) *Eine Funktion* f *von einer Menge* A *in eine Menge* B *wird notiert als* $f : A \to B$*.*

1. *Mit* $ran(f)$ *bezeichnen wir den Wertebereich einer Funktion* f*, d.h.* $W(f) = \{f(a) \mid a \in A\} = \{b \mid b \in B \text{ und } b = g(a) \text{ für ein } a \in A\}$*. Dabei steht ran abkürzend für den anglischen Begriff* range.

2. $f : A \to B$ *heißt* injektive*, falls für alle* $a_1, a_2 \in A$ *aus* $f(a_1) = f(a_2)$ *folgt* $a_1 = a_2$*.*

3. $f : A \to B$ *heißt* surjektive*, falls* $ran(f) = B$*.*

4. *Sind zwei Funktionen* $f : A \to B$ *und* $g : B \to C$ *gegeben, so bezeichnet* $f \circ g$ *die Hintereinanderausführung von zuerst* f *und dann* g*.*

Kapitel 2

Mehrwertige Logiken

2.1 Einführung in die mehrwertige Logik

2.1.1 Historie

Die Anfänge der formalen mehrwertigen Logik sind in erste Linie mit den Namen Jan Łukasiewicz, Emil Post und S.C. Kleene verbunden, die unabhängig voneinander ihre auch heute noch häufig zitierten Pionierarbeiten veröffentlich haben. Die Arbeit [Łukasiewicz, 1920] war durch philosophische Überlegungen motiviert, z.B. durch die Frage, welcher Wahrheitswert einer Aussage jetzt zukommt, deren Wahrheit oder Falschheit erst in der Zukunft offenbar wird. Man denke hier an Aussagen der Art „Am nächsten Sonntag wird es regnen“. Der kurze Artikel [Łukasiewicz, 1920] ist in polnischer Sprache geschrieben. Zehn Jahre später erschien der Aufsatz [Łukasiewicz, 1930] in deutscher Sprache. Wenn man tatsächlich die Quellen der Logik von Łukasiewicz lesen möchte, so empfiehlt sich die englische Übersetzung dieses Aufsatzes in der Sammlung [S.McCall, 1967]. Die Motivation für die Arbeit [Post, 1921] von Emil Post war hauptsächlich die Untersuchung algebraischer Eigenschaften einer gewissen Klasse von Algebren. In Kleenes Papier [Kleene, 1938] schließlich wird eine dreiwertige Logik aus puren Zweckmäßigkeitsgründen eingeführt, um möglichst elegant mit partiell rekursiven Funktionen umgehen zu können. Das Hilfsmittel der dreiwertigen Logik ist so sehr dem Zweck untergeordnet, daß im Titel der Arbeit kein Hinweis darauf zu finden ist. Eine detailiertere Darstellung der Geschichte der mehrwertigen Logik findet man z.B. in [Bolc & Borowik, 1992] und [S.McCall, 1967]. Eine erste Monographie erschien mit dem Buch [J. B. Rosser, 1952] und eine erste umfassende Übersicht über eine Vielzahl mehrwertiger Logiksysteme mit [Rescher, 1989]. Beide Bücher sind heute nur noch von historischem Interesse. Bei der Vorbereitung dieses Kapitels wurde hauptsächlich auf die folgenden Übersichtsarbeiten zurückgegriffen:

- das Kapitel über mehrwertige Logik von Alasdair Urquhart in Band III des Handbook of Philosophical Logic, [Urquhart, 1986],
- das Kapitel von Siegfried Gottwald in dem Sammelband „Einführung in die Nichklassische Logik“ des Akademie-Verlags, Berlin, [Gottwald, 1990],
- das Buch Many-Valued Logics, der erste bisher erschienene Band eines

auf zwei Bände ausgelegten Werkes von Bolc und Borowik, [Bolc & Borowik, 1992],

- das Buch von Reiner Hähnle, [Hähnle, 1993],
- Grzegorz Malinowski [Malinowski, 1993].
- Siegfried Gottwald [Gottwald, 2001] ist das aktuellste und umfassendste Buch zur mehrwertigen Logik.

2.1.2 Partielle Logik

Bevor wir mit der Präsentation eines ersten Beispiels einer mehrwertigen Logik beginnen, wollen wir zunächst die Frage stellen, wie überhaupt der Wunsch nach mehr als zwei Wahrheitswerten entstand. Auf diese Frage gibt es natürlich mehrere Antworten, je nachdem in welchem Bereich man sie stellt. Die folgenden Überlegungen stammen aus dem Bereich der Linguistik, genauer aus dem Teilbereich der Linguistik, der versucht die Bedeutung natürlichesprachiger Sätze formal zu fassen. Beginnen wir mit dem folgenden Text:

> Da die *Earl Camden* erst in einigen Tagen auslief, saß John Franklin im Hafen von Whampoa neben dem Maler William Westall untätig auf einer Mauer und beobachtete, was verladen wurde. Schiffe von über acht Fuß Tiefgang durften nicht flußaufwärts bin nach Kanton.…
> aus [Nadolny, 1987]

Schon die Erfassung der Bedeutung dieses kleinen Stückchens Text bietet enorme Schwierigkeiten. Wir wollen hier nur auf einen Aspekt, vielleicht ist es der allereinfachste, eingehen. Wir wollen die Basisfakten B aus dem Text herausziehen. Dazu müssen wir als erstes ein Basisvokabular festlegen. Dazu gehören sicherlich Bezeichnungen für die beiden genannten Personen JF und WW. Weiter erscheint es nützlich Zeichen für die erwähnte Mauer M_1 und die Orte H (für $Hafen\_von\_Whampoa$) und $Kanton$ zur Verfügung zu haben. Als Prädikate brauchen wir in jedem Fall $sitzt(x)$,$beobachtet(x)$,$auf(x, y)$,in(x,y). Damit kann man schon einmal notieren:

$$sitzt(JF), beobachtet(JF), auf(JF, M_1), in(JF, H)$$

Dabei ist nicht berücksichtigt, daß die beschriebene Situation in der Vergangenheit liegt. Aber das gehört nicht nicht mehr zu den Basisfakten und könnte später korrigiert werden indem die ganze Situationsbeschreibung durch einen modalen Operator für die Vergangenheit modifiziert wird. Naheliegender ist nach dem Status der Elementaraussagen

$$sitzt(WW), auf(WW, M_1), in(WW, H), beobachtet(WW)$$

zu fragen. Die ersten drei Aussagen können wir als zutreffend aus dem Text erschließen, über die letzte Aussage wissen wir nichts und höchstwahrscheinlich, ist es für den Fortgang der Geschichte auch unerheblich. Also lassen wir sie aus der Liste der bekannten Fakten weg.

$$sitzt(WW), auf(WW, M_1), in(WW, H)$$

Das wirft nun aber ein ernstes Problem auf. Die Aussagen $in(JF, Kanton)$ und $in(WW, Kanton)$ sind auch nicht aufgeführt, aber aus einem anderen Grund: wir gehen davon aus, daß sie nicht zutreffen. In der **partiellen Logik** löst man dieses Problem dadurch, daß zur Beschreibung einer Situation eine Wahrheitsfunktion $v : B \rightarrow \{0, 1\}$ angegeben wird. In unserem Beispiel etwa

A	$v(A)$	A	$v(A)$
$sitzt(JF)$	1	$sitzt(WW)$	1
$beobachtet(JF)$	1	$beobachtet(WW)$	
$auf(JF, M_1)$	1	$auf(WW, M_1)$	1
$in(JF, H)$	1	$in(WW, H)$	1
$in(JF, Kanton)$	0	$in(WW, Kanton)$	0

Der einzige Unterschied in der partiellen Logik gegenüber der klassischen Logik ist die Abschwächung der Anforderungen an die Bewertungsfunktion v: sie muß nicht mehr eine totale Funktion sein. Das ist genau, was man zur Beschreibung der Bedeutung eines Textausschnitts braucht. Die positiven Fakten, die negativen Fakten und keine Angaben über die Elementaraussagen, über die man nichts weiß.

Wie werden in der partiellen Logik aussagenlogische Verknüfungen definiert? Der Ausgangspunkt der Überlegungen die schließlich zu einer Antwort auf diese Frage führen ist, daß letztlich die Situationen mit einer totalen Bewertungsfunktion v interessieren, auch wenn sie nicht immer zur Verfügung stehen. Sie sind der Idealzustand, an dem man sich orientiert. Der Begriff der

$\wedge$	1	$undef$	0
1	1	$undef$	0
$undef$	$undef$	$undef$	0
0	0	0	0

$\vee$	1	$undef$	0
1	1	1	1
$undef$	1	$undef$	$undef$
0	1	$undef$	0

Abbildung 2.1: Wahrheitstafeln für die partielle Logik

Erweiterung einer Bewertungsfunktion $v : B \rightarrow \{1, 0\}$ zu einer Bewertungsfunktion $v_1 : B \rightarrow \{1, 0\}$ wird jetzt wichtig. Wir schreiben dafür $v \sqsubseteq v_1$ und meinen damit, daß v_1 eine totale Funktion ist und mit v übereinstimmt an allen Stellen, an denen schon v definiert ist. Man definiert jetzt zum Beispiel für die Konjunktion $\wedge$:

$$v(A \wedge B) = 0 \text{ gdw } \text{ für alle } v_1 \text{ mit } v \sqsubseteq v_1 \text{ gilt } v_1(A \wedge B) = 0$$

und entsprechend für den zweiten Wahrheitswert:

$$v(A \wedge B) = 1 \text{ gdw } \text{ für alle } v_1 \text{ mit } v \sqsubseteq v_1 \text{ gilt } v_1(A \wedge B) = 1$$

Zusammengefasst ist das Ergebnis in Abb. 2.1 zu sehen. Das bemerkenswerte an diesem Ansatz ist, daß die Abweichung von der klassischen Logik zunächst sehr gering ist und aus einem einsichtigen Grund heraus erfolgt. Beim weiteren Arbeiten stellt es sich dann als nützlich heraus das Attribut $undefiniert$ durch ein kürzeres Symbol u zu ersetzen. In den Wahrheitstabellen tritt dieses Symbol dann auf und wird wie ein zusätzlicher Wahrheitswert behandelt ohne große Diskussion über seinen ontologischen Status. Wer sich für dieses Thema und insbesondere für die damit verbundenen linguistischen Fragen interessiert, den verweisen wir auf die Literatur [Blamey, 1986],[Fenstad *et al.*, 1987],[Langholm, 1996].

2.1.3 Eine dreiwertige Logik

Welche Bestandteile machen eine m-wertige Logik aus? Zuerst wird durch die Zahl $m, m \geq 2$, die Anzahl der Wahrheitswerte festgelegt. Wie diese bezeichnet werden, ist unerheblich. Wir einigen uns darauf, die Zahlen $0, \ldots, m-1$ zur Bezeichnung der m Wahrheitswerte zu benutzen, und setzen $M = \{0, \ldots, m-1\}$. Als nächstes muß festgelegt werden, welche Operatoren in der Logik als Basisoperatoren vorkommen sollen. Die Menge dieser Operatoren bezeichnen wir mit F. Insbesondere können in F auch Konstanten, betrachtet als 0-stellige Operatoren, vorkommen. Ebenso n-stellige Operatoren für $n \geq 3$, auch wenn wir dafür bisher keine Beispiele kennengelernt haben. Die Menge der Formeln der zu beschreibenden Logik ist die Menge aller Terme, die mit den Operatoren in F und einer Menge P von aussagenlogischen Variablen (= 0-stellige Prädikatszeichen) aufgebaut werden können. Als nächstes muß den Operatoren aus F eine Bedeutung beigelegt werden. Das ist z. B. durch die Angabe von Wahrheitstabellen möglich. Aber das ist nur eine Methode, einem n-stelligen Operationszeichen f eine n-stellige Funktion von M^n nach M zuzuordnen. Wir wollen deshalb nur voraussetzen, daß jedem f aus F eine Funktion $f^{\mathcal{M}}$ derselben Stelligkeit auf der Menge M zugeordnet wird. Die Struktur $\mathcal{M} = (M, f^{\mathcal{M}} : f \in F)$ nennt man eine F-Algebra. Im Kontext m-wertiger Logik wird $\mathcal{M}$ häufig auch eine Matrix genannt. Ist t eine aussagenlogische Formel über dem Vokabular F und v eine Abbildung, die jeder aussagenlogischen Variable in t einen Wahrheitswert in M zuordnet, so läßt sich durch Auswertung des Terms t in der Algebra $\mathcal{M}$ der Wahrheitswert $v_{\mathcal{M}}(t)$ berechnen. Als letztes Bestimmungsstück der Logik ist eine nichtleere Teilmenge $D \subseteq M$ anzugeben, die Menge der designierten Wahrheitswerte. Eine Formel t heißt wahr in der Matrix $\mathcal{M}$ unter der Belegung v, falls $v_{\mathcal{M}}(t) \in D$ gilt. t heißt eine $\mathcal{M}$-Tautologie, falls für alle v gilt $v_{\mathcal{M}}(t) \in D$.

Als ein erstes Beispiel für eine mehrwertige Logik betrachten wir $\mathcal{L}_3$. Die Syntax der dreiwertigen Logik $\mathcal{L}_3$, die wir in diesem Abschnitt einführen wollen, unterscheidet sich von derjenigen des normalen, zweiwertigen Prädikatenkalküls allein dadurch, daß in $\mathcal{L}_3$ neben dem Negationszeichen $\neg$ noch ein zweites Negationszeichen $\sim$ vorhanden ist. Zur Unterscheidung nennen wir $\neg$ die starke und $\sim$ die schwache Negation. Die Formeln von $\mathcal{L}_3$ über einem Vokabular V werden wie gewohnt aus den Grundsymbolen aufgebaut. Die Menge aller $\mathcal{L}_3$-Formeln wird mit $Fml_3(V)$ bezeichnet. Wir werden gelegentlich allein den aussagenlogischen Teil von $\mathcal{L}_3$ betrachten wollen. Am einfachsten läßt sich der aussagenlogische Fall unter den Fall allgemeiner Formeln einordnen, wenn wir, ebenso wie wir schon null-stellige Funktionszeichen (= Konstantenzeichen) erlaubt haben, auch null-stellige Prädikatszeichen erlauben. Diese null-stelligen Prädikatszeichen nennen wir auch Aussagenvariablen. Eine aussagenlogische $\mathcal{L}_3$-Formel ist danach eine Formel in Fml_3, die keine Quantoren und nur null-stellige Prädikatszeichen enthält.

$\wedge$	1	u	0
1	1	u	0
u	u	u	0
0	0	0	0

$\vee$	1	u	0
1	1	1	1
u	1	u	u
0	1	u	0

Abbildung 2.2: Wahrheitstafeln für $\wedge$ und $\vee$

Eine $\mathcal{L}_3$-Struktur im Vokabular V, bezeichnet mit $\mathcal{M} =< \mathcal{M}_0, v_\mathcal{M} >$, besteht aus einer Prästruktur $\mathcal{M}_0$ für V, d.h. aus einer Menge zusammen mit der üblichen Interpretation für Konstanten und Funktionssymbole, und einer Funktion $v_\mathcal{M}$, die jedem Relationszeichen P und jedem n-Tupel $m_1, \ldots, m_n$ von Elementen aus $\mathcal{M}_0$ (n = Stellenzahl von P) einen der Wahrheitswerte 1, u oder 0 zuordnet.

$$\begin{array}{lcll} v_\mathcal{M}(P(m_1, \ldots, m_n)) & = & 1 & (wahr) \\ v_\mathcal{M}(P(m_1, \ldots, m_n)) & = & u & (unbestimmt) \\ v_\mathcal{M}(P(m_1, \ldots, m_n)) & = & 0 & (falsch) \end{array}$$

Die Bewertungsfunktion $v_\mathcal{M}$ läßt sich auf alle $\mathcal{L}_3$-Formeln fortsetzen. Dabei berechnet sich $v_\mathcal{M}(A \wedge B)$ und $v_\mathcal{M}(A \vee B)$ mit Hilfe der Tabelle in Abbildung 2.2 aus $v_\mathcal{M}(A)$ und $v_\mathcal{M}(B)$.

Entsprechendes gilt für die übrigen aussagenlogischen Junktoren.

Die in der Abbildung 2.2 gegebenen Regeln für $\wedge$ und $\vee$ kann man einfach zusammenfassen zu

$$\begin{array}{lcl} v(A \wedge B) & = & \text{das Minimum von } v(A) \text{ und } v(B) \\ v(A \vee B) & = & \text{das Maximum von } v(A) \text{ und } v(B), \end{array}$$

wenn man sich die Wahrheitswerte in der Reihe $0 < u < 1$ angeordnet denkt.

Es mag für das Verständnis der Wahrheitstafeln für die Negationen hilfreich sein, die folgende natürlichsprachliche Umschreibung im Kopf zu haben:

A	$\neg A$	$\sim A$	$\sim \neg A$	$\sim\sim A$	$\neg\neg A$	$\neg \sim A$
1	0	0	1	1	1	1
u	u	1	1	0	u	0
0	1	1	0	0	0	0

Abbildung 2.3: Wahrheitstafeln für Negationen

$v(A) = 1$	gdw.	A ist wahr
$v(\neg A) = 1$	gdw.	A ist falsch
$v(\sim A) = 1$	gdw.	A ist nicht wahr
$v(\sim \neg A) = 1$	gdw.	A ist nicht falsch.

Die angegebenen Wahrheitstafeln für die aussagenlogischen Verknüpfungen sind nicht die einzig möglichen, und man findet in der Tat in der Literatur zahlreiche Varianten. Für definite Argumente 0 und 1 stimmen die Wahrheitstafeln mit der zweiwertigen Logik überein. Dieser Teil ist unumstritten. Dagegen ist nicht so klar, welchen Wert $0 \wedge u$ haben soll. In unserem Fall haben wir $0 \wedge u = 0$ gesetzt, was man mit dem Hinweis begründen könnte, daß unabhängig davon, ob man u durch 0 oder 1 ersetzt, 0 als Ergebnis erhält. Läßt man sich dagegen von dem Grundsatz leiten, daß ein Ausdruck nur dann einen von u verschiedenen Wert erhalten soll, wenn alle Teilausdrücke einen von u verschiedenen Wert haben, so wird man zu der Festsetzung $0 \wedge_0 u = u$ geführt. Wir sind der Meinung, daß eine Diskussion über die *richtige* Definition der dreiwertigen Wahrheitstafeln nur im Rahmen einer konkreten Anwendung sinnvoll geführt werden kann. Für uns ist die Bereitstellung eines universellen Rahmens das vorrangige Ziel, und wir sehen das vorgelegte formale System durch die Aussagen von Lemma 4 auf Seite 39 hinreichend gerechtfertigt.

Es bleibt noch in der induktiven Definition von $v_{\mathcal{M}}$ der Induktionsschritt für die Quantoren zu erklären.

Definition 11

Wir nehmen für den Augenblick an, daß für jedes Element m aus der betrachteten Struktur $\mathcal{M}$ eine gleichnamige Konstante im Vokabular V vorhanden ist, daß somit für jedes $m \in M$ = Universum von $\mathcal{M}$ und jede Formel $A(x)$ aus $Fml_3(V)$ auch $A(m/x)$ wieder eine $\mathcal{L}_3$-Formel ist. Diese Annahme ist sicherlich harmlos, erspart uns aber die zusätzliche Angabe einer Belegungs-

funktion b für die freien Variablen der Formel A. Wir schreiben außerdem der Einfachheit halber v anstelle von $v_{\mathcal{M}}$.

$$\begin{array}{lll}
v(\forall x A(x)) = 1 & gdw. & \textit{für alle } m \in \mathcal{M} \textit{ gilt } v(A(m/x)) = 1. \\
v(\forall x A(x)) = 0 & gdw. & \textit{es gibt ein } m \in \mathcal{M}, \textit{ so daß gilt } v(A(m/x)) = 0. \\
v(\forall x A(x)) = u & & \textit{in allen anderen Fällen.} \\
v(\exists x A(x)) = 1 & gdw. & \textit{es gibt ein } m \in \mathcal{M}, \textit{ so daß gilt } v(A(m/x)) = 1. \\
v(\exists x A(x)) = 0 & gdw. & \textit{für alle } m \in \mathcal{M} \textit{ gilt } v(A(m/x)) = 0. \\
v(\exists x A(x)) = u & & \textit{in allen anderen Fällen.}
\end{array}$$

Besonders in Fällen, in denen anstelle von $\mathcal{M}$ *eine durch eine komplexere Notation beschriebene Struktur tritt, schreiben wir* $v(\mathcal{M}, A) = w$ *anstelle von* $v_{\mathcal{M}}(A) = w$. *In der obigen Definition haben wir um die Notation zu vereinfachen Variablenbelegungen ersetzt durch Konstanten für Elemente. Wenn wir stattdessen eine Variablenbelegung* $\beta : Var \to M$ *benutzen, schreiben wir* $v_{\mathcal{M},\beta}(A)$ *anstelle von* $v_{\mathcal{M}}(A)$.

Definition 12
Sei S eine Menge von $Fml_3(V)$*-Formeln und A eine* $Fml_3(V)$ *-Formel ohne freie Variablen.*
Wir nennen A eine dreiwertige logische Folgerung aus S, wenn für alle dreiwertigen Strukturen $\mathcal{M}$ *zum Vokabular V, für die* $v_{\mathcal{M}}(B) = 1$ *für alle* $B \in S$ *gilt, auch* $v_{\mathcal{M}}(A) = 1$ *gilt. Wir schreiben dafür* $S \vdash_3 A$.
Die Formel A heißt eine dreiwertige Tautologie oder allgemeingültig, wenn $v_{\mathcal{M}}(A) = 1$ *gilt für alle dreiwertigen Strukturen* $\mathcal{M}$ *Wir schreiben dafür* $\vdash_3 A$.

Die aussagenlogischen Verknüpfungen $\supset$ und $\equiv$ werden als definierte Zeichen eingeführt, und zwar:

$$\begin{array}{lcll}
\text{schwache Implikation} & A \supset B & := & \sim A \vee B \\
\text{schwache Äquivalenz} & A \equiv B & := & (A \supset B) \wedge (B \supset A)
\end{array}$$

Durch die Anwesenheit zweier Negationszeichen gibt es auch zwei Möglichkeiten, die Implikation und die Äquivalenz aus den gegebenen Verknüpfungen zu definieren. Die oben gewählte Möglichkeit führt zur schwachen Implikation und zur schwachen Äquivalenz.

Zur Bequemlichkeit des Lesers drucken wir auch die Wahrheitstafeln für diese beiden Verknüpfungen in Abbildung 2.4 ab.

A ⊃ B			
B ↓ A →	1	u	0
1	1	1	1
u	u	1	1
0	0	1	1

A ≡ B			
	1	u	0
1	1	u	0
u	u	1	1
0	0	1	1

Abbildung 2.4: Wahrheitstafeln für ⊃ und ≡

A → B			
B ↓ A →	1	u	0
1	1	1	1
u	u	u	1
0	0	u	1

A ↔ B			
	1	u	0
1	1	u	0
u	u	u	u
0	0	u	1

Abbildung 2.5: Wahrheitstafeln für → und ↔

Als weitere definierte, aussagenlogische Junktoren wollen wir die starke Implikation und die starke Äquivalenz erwähnen, definiert durch:

$$\begin{aligned} A \rightarrow B &:= \neg A \vee B \\ A \leftrightarrow A &:= (A \rightarrow B) \wedge (B \rightarrow A) \end{aligned}$$

Wahrheitstafeln für die starke Implikation und die starke Äquivalenz sind in Abbildung 2.5 wiedergegeben. Für die aussagenlogische Äquivalenz ergeben sich im dreiwertigen Rahmen besonders viele Varianten. Wir fassen die wichtigsten zusammen, für die zugehörigen Wahrheitstabellen siehe Abbildung 2.6. Aus diesen Tabellen ist zum Beispiel ersichtlich, daß $\leftrightarrow$ und $\Leftrightarrow$ denselben Wahrheitswerteverlauf haben.

$$\begin{array}{lcl}
A \supset B & := & \sim A \vee B \\
A \rightarrow B & := & \neg A \vee B \\
A \equiv B & := & (A \supset B) \wedge (B \supset A) \\
A \leftrightarrow B & := & (A \rightarrow B) \wedge (B \rightarrow A) \\
A \approx B & := & (A \equiv B) \wedge (\neg A \equiv \neg B) \\
A \Leftrightarrow B & := & (A \leftrightarrow B) \wedge (\neg A \leftrightarrow \neg B) \\
A \;\; id \;\; B & := & \sim\sim (A \approx B)
\end{array}$$

A	B	$A \equiv B$	$A \leftrightarrow B$	$A \approx B$	$A \Leftrightarrow B$	$A \;\; id \;\; B$
0	0	1	1	1	1	1
0	u	1	u	u	u	0
0	1	0	0	0	0	0
u	0	1	u	u	u	0
u	u	1	u	1	u	1
u	1	u	u	u	u	0
1	0	0	0	0	0	0
1	u	u	u	u	u	0
1	1	1	1	1	1	1

Abbildung 2.6: Äquivalenzen

Das nächste Lemma enthält eine Liste aus der klassischen Logik bekannter allgemeingültiger Äquivalenzen, die auch in der Logik $\mathcal{L}_3$ in der stärksten Form (mit *id* als Äquivalenzrelation) gültig sind.

Lemma 1

1. $\neg\neg A \; id \; A$

2. $\sim\sim A \;\equiv\; A$

3. $\neg \sim A \;\equiv\; A$

4. $(A \wedge B) \vee C \; id \; (A \vee C) \wedge (B \vee C)$

5. $(A \vee B) \wedge C \; id \; (A \wedge C) \vee (B \wedge C)$

6. $\neg(A \vee B) \; id \; \neg A \wedge \neg B$

7. $\neg(A \wedge B)$ *id* $\neg A \vee \neg B$

8. $\sim (A \vee B)$ *id* $\sim (A) \wedge \sim (B)$

9. $\sim (A \wedge B)$ *id* $\sim (A) \vee \sim (B)$

10. $\sim (\forall x A)$ *id* $\exists x \sim A$

11. $\neg(\forall x A)$ *id* $\exists x \neg A$

12. $\sim (\exists x A)$ *id* $\forall x \sim A$

13. $\neg(\exists x A)$ *id* $\forall x \neg A$

Beweis: Übungsaufgabe.

Lemma 2 *Die folgenden $\mathcal{L}_3$-Formeln sind Tautologien:*

1. $A \supset A$

2. $A \supset (B \supset A)$

3. $(A \supset B) \supset ((B \supset C) \supset (A \supset C))$

4. $(A \leftrightarrow B)$ *id* $(\neg A \leftrightarrow \neg B)$

Die folgenden $\mathcal{L}_3$-Formeln sind keine Tautologien:

1. $A \rightarrow A$

2. $A \rightarrow (B \rightarrow A)$

3. $(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))$

4. $(A \equiv B)$ *id* $(\neg A \equiv \neg B)$

5. $(A \equiv B)$ *id* $(\sim A \equiv \sim B)$

6. $(A \leftrightarrow B)$ *id* $(\sim A \leftrightarrow \sim B)$

Lemma 3 *Sei A eine $\mathcal{L}_3$-Formel, die das starke Negationszeichen $\neg$ nicht enthält. Dann gilt*

A ist eine $\mathcal{L}_3$-Tautologie
genau dann, wenn
A ist eine zweiwertige Tautologie.

Wir nehmen dabei an, daß das im zweiwertigen Fall einzige vorhandene Negationszeichen mit dem schwachen Negationszeichen $\sim$ identifiziert wird.

Beweis: Trivialerweise ist jede $\mathcal{L}_3$-Tautologie auch eine zweiwertige Tautologie. Für jede $\mathcal{L}_3$-Struktur $\mathcal{M} =< \mathcal{M}_0, v_0 >$ sei $\mathcal{M}_1 =< \mathcal{M}_0, v_1 >$, wobei v_1 mit v_0 übereinstimmt, mit der Ausnahme, daß für jedes Argument, für welches v_0 den Wert u hatte, nun v_1 den Wert 0 hat. Offensichlich ist A eine zweiwertige Tautologie genau dann, wenn für alle dreiwertigen Strukturen $\mathcal{M} =< \mathcal{M}_0, v_0 >$ gilt $v_1(A) = 1$. Wir behaupten, daß für alle Formeln A und alle dreiwertigen Strukturen $\mathcal{M}$ gilt:

Aus	$v_0(A) = 1$	folgt	$v_1(A) = 1$
Aus	$v_0(A) = 0$	folgt	$v_1(A) = 0$
Aus	$v_0(A) = u$	folgt	$v_1(A) = 0$

Diese Behauptung läßt sich leicht durch Induktion über die Komplexität von A beweisen.
Auf Seite 72 kommen wir noch einmal auf dieses Lemma zurück und skizzieren einen alternativen Beweis.

∎

Die Formel $\neg A \vee A$ ist keine $\mathcal{L}_3$-Tautologie. Lemma 3 ist also nicht wahr, wenn $\neg$ durch $\sim$ ersetzt wird.

2.1.4 Übungsaufgaben

Übungsaufgabe 2.1.1 *Sei A eine aussagenlogische Formel in Fml_3 (mit den Junktoren $\sim$, $\neg$, $\wedge$ und $\vee$) mit n aussagenlogischen Variablen, dann ist für die durch A induzierte Funktion $f_A : \{1, u, 0\}^n \to \{1, u, 0\}$ stets $f_A(1, 1, \ldots, 1) \neq u$.*

Übungsaufgabe 2.1.2 *Ist A eine $\mathcal{L}_3$-Formel, in der $\sim$ nicht vorkommt, dann ist A keine Tautologie.*

A $\Rightarrow$ B			
A $\downarrow$ B $\rightarrow$	1	u	0
1	1	u	0
u	1	1	u
0	1	1	1

Abbildung 2.7: Wahrheitstabelle für $\Rightarrow$

Übungsaufgabe 2.1.3

1. *Zeigen Sie, daß* $\forall x q(x) \supset q(t/x)$ *für jeden Term* t *eine* $\mathcal{L}_3$*-Tautologie ist.*

2. *Finden Sie ein Gegenbeispiel, das zeigt, daß* $\forall x q(x) \rightarrow q(t/x)$ *keine Tautologie ist.*

3. *Die Implikation* $\Rightarrow$ *werde durch die Wahrheitstabelle in Abbildung 2.7 gegeben. Es handelt sich dabei um die Implikation von Łukasiewicz, der wir später noch einmal in Abbildung 2.12 begegnen werden. Die Implikation* $\Rightarrow$ *ist somit stärker als* $\supset$*, aber schwächer als* $\rightarrow$*.*
   Zeigen Sie, daß $\forall x q(x) \Rightarrow q(t/x)$ *eine Tautologie ist.*

Übungsaufgabe 2.1.4 *Ist* $A \supset B$ *eine Tautologie, dann auch* $A \wedge C \supset B \wedge C$ *für jedes* C*.*

Übungsaufgabe 2.1.5 *Welche der folgenden Aussagen sind richtig?*

1. *Wenn* $A \equiv B$ *eine Tautologie ist und* A *eine Tautologie ist, dann ist auch* B *eine Tautologie.*

2. *Wenn* $A \equiv B$ *eine Tautologie ist und* A *erfüllbar ist, dann ist auch* B *erfüllbar.*

3. *Wenn* $A \equiv B$ *eine Tautologie ist und* A *eine Nichttautologie ist, dann ist auch* B *eine Nichttautologie.*

4. *Wenn* $A \equiv B$ *eine Tautologie ist und* A *zweiwertig ist, dann ist auch* B *zweiwertig.*

Dabei heißt eine Formel A eine **Nichttautologie***, wenn für jede dreiwertige Struktur* $\mathcal{M} =< \mathcal{M}_0, v_{\mathcal{M}} >$ *gilt* $v_{\mathcal{M}} \neq 1$. *A heißt* **zweiwertig***, wenn für jede dreiwertige Struktur* $\mathcal{M} =< \mathcal{M}_0, v_{\mathcal{M}} >$ *gilt* $v_{\mathcal{M}}(A) \in \{1, 0\}$.

Übungsaufgabe 2.1.6 *Gilt das Deduktionstheorem für die schwache Implikation, für die starke Implikation? Genauer:*

1. *Ist* $A \vdash_3 B$ *äquivalent zu* $\vdash_3 A \supset B$*?*
2. *Ist* $A \vdash_3 B$ *äquivalent zu* $\vdash_3 A \to B$*?*

Übungsaufgabe 2.1.7 *Wir haben gesehen, daß es viele Möglichkeiten gibt, die aus der zweiwertigen Logik bekannten Operatoren zu dreiwertige Operatoren zu erweitern. Besonders augenfällig ist das für die Implikation und für die logische Äquivalenz. Man fragt sich, ob es nicht vielleicht ein einleuchtendes Prinzip dafür gibt? Hier ist ein Vorschlag.*

Definition 13 (Standardfortsetzung)
Sei $f : \{0, 1\}^n \to \{0, 1\}$ *gegeben. Die Standardfortsetzung* $f^* : \{0, u, 1\}^n \to \{0, u, 1\}$ *von* f *wird wie folgt bestimmt. Sei* $\langle w_1, \ldots, w_n \rangle$ *ein Argumenttupel aus* $\{0, u, 1\}^n$. *Die Menge* $U(w_1, \ldots, w_n) \subseteq \{0, 1\}^n$ *entsteht, indem auf alle möglichen Arten die* w_i *mit* $w_i = u$ *durch* 0 *und* 1 *ersetzt werden. So gilt zum Beispiel* $U(u, 1, u) = \{\langle 0, 1, 0 \rangle, \langle 0, 1, 1 \rangle, \langle 1, 1, 0 \rangle, \langle 1, 1, 1 \rangle\}$.

$$f^*(w_1, \ldots, w_n) = \begin{cases} 0 & \text{falls für alle } \vec{v} \in U(w_1, \ldots, w_n) \text{ gilt } f(\vec{v}) = 0 \\ 1 & \text{falls für alle } \vec{v} \in U(w_1, \ldots, w_n) \text{ gilt } f(\vec{v}) = 1 \\ u & \text{sonst} \end{cases}$$

1. *Zeigen Sie, daß die dreiwertigen Operatoren* $\wedge, \vee, \neg, \to, \leftrightarrow$ *Standardfortsetzungen ihrer jeweiligen zweiwertigen Gegenstücke sind.*
2. *Zeigen Sie, daß das nicht zutrifft auf* $\supset$ *und* $\equiv$.
3. *Zeigen Sie, daß aus* $f(\vec{w}) = 1$ *für alle* $\vec{w} \in \{0, 1\}^n$ *auch für alle* $\vec{v} \in \{0, u, 1\}^n$ $f^*(\vec{v}) = 1$ *folgt.*
4. *Zeigen Sie, daß die Standardfortsetzung nicht immer mit der Hintereinanderausführung von Funktionen vertauschbar ist, d.h. finden Sie Funktionen* g_1, g_2*, so daß*
$$[g_1(g_2(\vec{w}))]^* \neq g_1^*(g_2^*(\vec{w}))$$

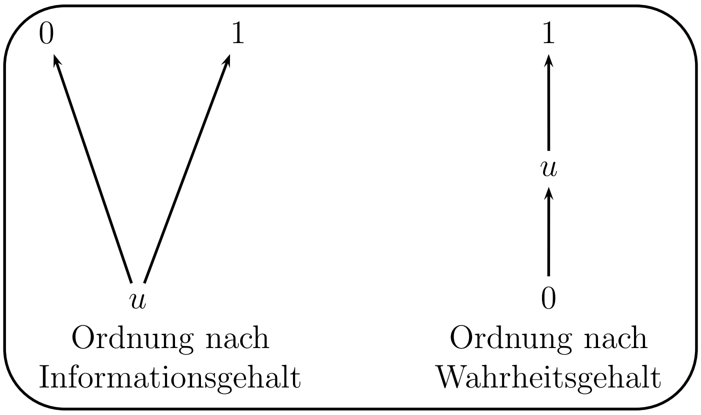


Abbildung 2.8: Ordnungen auf den Wahrheitswerten von $\mathcal{L}_3$

Übungsaufgabe 2.1.8 *Die Wahrheitswerte der Logik $\mathcal{L}_3$ lassen sich auf zwei verschiedene Weisen anordnen, einmal nach dem Wahrheitsgehalt (diese Ordnung führt zu $v(A \wedge B) = min\{v(A).v(B)\}$) und einmal nach dem Informationsgehalt, siehe Abbildung 2.8. Diese Ordnung wollen wir mit $\leq_i$ bzw. in ihrer strikten Form mit $<_i$ bezeichnen. Die Ordnung $\leq_i$ wird auf n-Tupel von Wahrheitswerten erweitert durch komponentenweisen Vergleich, d.h. $\langle w_1, \ldots, w_n \rangle \leq_i \langle u_1, \ldots, u_n \rangle$ gdw für alle $1 \leq j \leq n$ gilt $w_j \leq_i u_j$. Wir nennen dann, wie üblich, eine Funktion $g : \{0, u, 1\}^n \to \{0, u, 1\}$* **monoton**, *wenn für alle $\vec{w}, \vec{v} \in \{0, u, 1\}^n$ mit $\vec{w} \leq_i \vec{v}$ auch $g(\vec{w}) \leq_i g(\vec{v})$ gilt.*

1. *Zeigen Sie, daß für jedes $f : \{0, 1\} \to \{0, 1\}$ die Standardfortsetzung f^* monoton ist.*

Die umgekehrte Implikation ist so nicht richtig. Um dennoch eine hinreichende und notwendige Bedingung dafür zu erhalten, daß eine Funktion eine Standardfortsetzung ist, benötigen wir den Begriff der maximalen Monotonie.

Definition 14

1. *$g_1 \leq g_2$ gdw für alle $\vec{w}$ gilt $g_1(\vec{w}) \leq g_2(\vec{w})$*
2. *Eine Funktion $g : \{0, u, 1\}^n \to \{0, u, 1\}$ heißt* **maximal monoton** *bzgl. $\leq$, wenn sie monoton bzgl. $\leq$ ist und keine Funktion g_1 mit $g < g_1$ monoton bzgl. $\leq$ ist.*

f	1	u	1
1	1	u	1
u	u	u	1
0	0	u	1

Abbildung 2.9: Wahrheitstafeln einer monotonen, nicht maximal monotonen Funktion

Abbildung 2.9 zeigt eine monotone, aber nicht maximal montone Funktion.

2. *Zeigen Sie, daß eine Funktion $g : \{0, u, 1\}^n \to \{0, u, 1\}$ genau dann die Standardfortsetzung einer Funktion $f : \{0, 1\}^n \to \{0, 1\}$ ist, wenn g maximal monton bezüglich der Ordnung nach $\leq_i$ ist.*

Übungsaufgabe 2.1.9

Definition 15
Für die Zwecke dieser Definition bezeichne $\sim$ die Relation auf der Menge $\{0, u, 1\}$, die durch

$$a \sim b \text{ gdw } a = b = 1 \text{ oder } (a \neq 1 \text{ und } b \neq 1)$$

gegeben ist. Die Relation $\sim$ wird für jedes n zu einer Relation auf $\{0, u, 1\}^n$ fortgesetzt durch komponentenweise Erweiterung, i.e.

$$\langle a_1 \ldots a_n \rangle \sim \langle b_1 \ldots b_n \rangle \text{ gdw für alle } 1 \leq i \leq n \text{ gilt } a_i \sim b_i.$$

Eine Funktion $f : \{0, u, 1\} \to \{0, u, 1\}$ heißt u-0-kongruent*, falls für alle $\vec{a}, \vec{b} \in \{0, u, 1\}^n$ gilt:*

$$\text{wenn } \vec{a} \sim \vec{b} \text{ dann } f(\vec{a}) \sim f(\vec{b})$$

Zeigen Sie, daß für jede Formel A aus $\mathcal{L}_3$, die das starke Negationszeichen $\neg$ nicht enthält, die durch A induzierte Funktion $f(A)$ u-0-kongruent ist.

Übungsaufgabe 2.1.10

Die im folgenden benutzte Ordnung $<_i$ nach Informationsgehalt auf der Menge der Wahrheitswerte $\{0, u, 1\}$ wurde auf Seite 34 eingeführt.

Definition 16 *Seien $\mathcal{M}$ und $\mathcal{N}$ dreiwertige Strukturen, $F : M \to N$ eine* **surjektive** *Abbildung, dann is $\mathcal{N}$ ist eine* **Abstraktion** *von $\mathcal{M}$ via F, in Zeichen $\mathcal{M} \sqsubseteq^F \mathcal{N}$, wenn*

1. *für alle n, alle n-stelligen Funktionszeichen f und alle $a_1, \ldots, a_n \in M$ gilt*
$$F(f^{\mathcal{M}}(a_1, \ldots, a_n)) = f^{\mathcal{N}}(F(a_1), \ldots, F(a_n))$$

2. *für alle n, alle n-stelligen Prädikatszeichen p und alle $b_1, \ldots, b_n \in M$ gilt:*
$$v_{\mathcal{N}}(p(F(b_1), \ldots, F(b_n))) \leq_i v_{\mathcal{M}}(p(b_1, \ldots, b_n))$$

Es ist auch verbreitet in dem vorliegenden Fall zu sagen: $\mathcal{M}$ **subsumiert** *die Struktur $\mathcal{N}$ bzgl F. Wir benutzen auch die Formulierungen $\mathcal{N}$* **abstrahiert** *$\mathcal{M}$, oder $\mathcal{N}$ ist eine F-Abstraktion von $\mathcal{M}$.*

Sei außerdem $\mathcal{L}$ eine Logik, in der alle aussagenlogischen Operatoren Standardfortsetzungen ihrer zweiwertigen Restriktion sind, siehe Definition 13 auf Seite 33. Die Interpretation der Quantoren in $\mathcal{L}$ sei wie in $\mathcal{L}_3$, Definition 11.

Zeigen Sie, *daß für jedes Paar von dreiwertigen Strukturen $\mathcal{M}$, $\mathcal{N}$ und jede surjektive Abbildung $F : M \to N$ mit $\mathcal{M} \sqsubseteq^F \mathcal{N}$, für jede Formel A aus $\mathcal{L}$ und jede Variablenbelegung $\beta : Var \to M$ gilt*

$$v_{\mathcal{N},\beta\circ F}(A) \leq_i v_{\mathcal{M},\beta}(A)$$

Das ist die Essenz von [Sagiv et al.*, 2000, Theorem 3.11, p.15] und [Sagiv* et al.*, 2001, Theorem 4.9, p.71]. Das zitierte Resultat ist einerseits schwächer, da es für eine bestimmte dreiwertige Logik formuliert ist, auf der anderen Seite ist die Sprache etwas stärker (es gibt einen Operator für den transitiven Abschluß und so genannte* summary predicates*.*

Wir bringen noch einige Informationen darüber wie in den Arbeiten [Sagiv et al.*, 2000, Sagiv* et al.*, 2001] das in dieser Aufgabe zu beweisende Resultat eingesetzt wird.*

Sei dazu $\mathcal{M}$ eine prädikatenlogische, typischerweise zwei-wertige, Struktur. Man möchte feststellen, ob eine prädikatenlogische Formel A in $\mathcal{M}$ wahr ist. Als ein Beispiel soll die in Abbildung 2.10 beschriebene Sruktur $\mathcal{M}_{list}$ dienen.

elem.	px	py
u_1	1	0
u_2	0	0
u_3	0	0
u_4	0	0

$n(\_\,,\_)$	u_1	u_2	u_3	u_4
u_1	0	1	0	0
u_2	0	0	1	0
u_3	0	0	0	1
u_4	0	0	0	0

Abbildung 2.10: Beispielstruktur $\mathcal{M}_{list}$

Man möchte z.B. die Frage beantworten ob $(\mathcal{M}_{list}, \beta) \models \exists v_1 \exists v_2 (n(v_1, v) \wedge n(v_2, v) \wedge v_1 \neq v_2)$ *zutrifft für* $\beta(v) = u_2$.

Sei E *eine beliebige Kongruenzrelation auf* M *und* $N = \{[m]_E \mid m \in M\}$ *die Menge der Kongruenzklassen bezgl.* E*. Wir definieren eine dreiwertige Struktur* $\mathcal{N}$ *auf* N *mit den Wahrheitswerten* $\{0, \frac{1}{2}, 1\}$*, indem wir die Funktionszeichen* f*, wie in Quotientenstrukturen üblich definieren, d.h.*

$$f^{\mathcal{N}}([m_1]_E, \ldots, [m_n]_E) = [f^{\mathcal{M}}(m_1, \ldots, m_n)]_E$$

Sei p *ein* n*-stelliges Pädikatensymbol, dann setzen wir*

$$p^{\mathcal{N}}([m_1]_E, \ldots, [m_n]_E) = \begin{cases} 0 & \text{falls für alle } a_1 \in [m_1]_E, \ldots, a_n \in [m_n]_E \text{ gilt} \\ & p^{\mathcal{M}}(a_1, \ldots, a_n) = 0 \\ 1 & \text{falls für alle } a_1 \in [m_1]_E, \ldots, a_n \in [m_n]_E \text{ gilt} \\ & p^{\mathcal{M}}(a_1, \ldots, a_n) = 1 \\ \frac{1}{2} & \text{sonst} \end{cases}$$

Man beachte, daß nach dieser Definition für den Randfall eines 0*-stelligen Prädikats* p *stets* $p^{\mathcal{N}} = p^{\mathcal{M}}$ *gilt.*

Sei F *die surjektive Abbildung von* $\mathcal{M}$ *auf* $\mathcal{N}$*, die jedem Element* $m \in M$ *seine* E*-Äquivalenzklasse zuordnet,* $F(m) = [m]_E$*, dann überprüft man leicht, daß* $\mathcal{M} \sqsubseteq^F \mathcal{N}$ *gilt. Nun wertet man die Formel* A *in* $\mathcal{N}$ *aus, was in der Regel einfacher ist, da* N *weniger Elemente besitzt als* M*. Falls die Auswertung* $v_{\mathcal{N}}(A)$ *einen definiten Wahrheitswert,* 0 *oder* 1*, ergibt, dann weiß man nach obigem Resultat, daß* A *auch in* $\mathcal{M}$ *falsch bzw. wahr ist. Aus* $v_{\mathcal{N}}(A) = u$ *kann man allerdings keine Rückschlüsse ziehen. Man kann dann mit einer anderen Kongruenzrelation* E_1 *das Verfahren noch einmal versuchen.*

In der Beispielstruktur $\mathcal{M}_{list}$ *gibt es keine Funktionszeichen, jede Äquivalenzrelatiion auf* M_{list} *ist also schon eine Kongruenzrelation. Man wählt hier* *geschickterweise die Relation*

$$E_{list}(m_1, m_2) \Leftrightarrow (px(m_1) \leftrightarrow px(m_2)) \text{ und } (py(m_1) \leftrightarrow py(m_2))$$

Dadurch entstehen zwei E_{list}-Äquivalenzklassen $n_1 = \{u_1\}$ und $n_2 = \{u_2, u_3, u_4\}$. Die zugehörige Struktur $\mathcal{N}_{list}$ is in Abbildung 2.11 zu sehen.

elem.	px	py
n_1	1	0
n_2	0	0

$n(\_\,,\_)$	n_1	n_2
n_1	0	$\frac{1}{2}$
n_2	0	$\frac{1}{2}$

Abbildung 2.11: Dreiwertige Beispielstruktur $\mathcal{N}_{list}$

Übungsaufgabe 2.1.11

Wie Wahrheitswerte bezeichnet werden ist ziemlich unerheblich. In dieser Aufgabe bezeichnen wir die Wahrheitswerte 1, 0, *u durch die Mengen* $\{1\}$, $\{0\}$, $\{1,0\}$.*Um auf diese Zuordnung explizit zugreifen zu können führen wir die Abbildung T ein mit $T(\{1\}) = 1$, $T(\{0\}) = 0$ und $T(\{1,0\}) = u$. Für eine Funktion $f : \{0,1\}^n \to \{0,1\}$ definieren wir eine Fortsetzung $f^+ : \{\{1\},\{0\},\{1,0\}\}^n \to \{\{1\},\{0\},\{1,0\}\}$ durch die* mengenwertige *Fortsetzung von f, i.e. $f^+(W_1, \ldots, W_n) = \{f(w_1, \ldots, w_n) \mid w_j \in W_j$ für alle $1 \leq j \leq n\}$*

Wir behaupten jetzt, daß die Funktion f^+ unter der durch T gegebenen Zuordnung mit f^* übereinstimmt, d.h.

$$t(f^+(W_1, \ldots, W_n)) = f^*(T(W_1), \ldots, T(W_n))$$

2.2 Funktionale Vollständigkeit

Jeder aussagenlogischen $\mathcal{L}_3$-Formel A mit n Aussagenvariablen läßt sich eine Funktion $f(A) : \{1, u, 0\}^n \rightarrow \{1, u, 0\}$ zuordnen, wobei für ein Argumenttupel $< w_1, \ldots, w_n >$ der Funktionswert $f(A)(w_1, \ldots, w_n)$ der Wahrheitswert der Formel A ist, wenn die aussagenlogischen Variablen mit den Wahrheitswerten $w_1, \ldots, w_n$ belegt sind.

Definition 17 (Funktionale Vollständigkeit) *Eine mehrwertige Logik $\mathcal{L}$ mit M als Menge der Wahrheitswerte heißt* **funktional vollständig**, *wenn zu jeder Funktion $g : M^n \rightarrow M$ eine aussagenlogische Formel A in $\mathcal{L}$ existiert mit $f(A) = g$.*

`noch offen, was mit n = 0 passieren soll`

Aufgabe 2.1.1 zeigt, daß $\mathcal{L}_3$ nicht funktional vollständig ist. Nach einer geringfügigen Modifikation der Logik $\mathcal{L}_3$ wird diese Aussage jedoch wahr. Die Logik $\mathcal{L}_3^+$ entsteht aus $\mathcal{L}_3$ durch Hinzunahme einer aussagenlogischen Konstanten u, die stets den Wahrheitswert u annimmt.

Lemma 4 *Die Logik $\mathcal{L}_3^+$ ist* **funktional vollständig**.

Beweis: Sei $g : \{1, u, 0\}^n \rightarrow \{1, u, 0\}$ gegeben. Die Konstruktion der gesuchten aussagenlogischen Formel A geschieht durch Induktion über n. Der Induktionsanfang $n = 0$ ist trivial. Hierbei wird die Existenz der aussagenlogischen Konstanten u benutzt. Im Induktionsschritt von $n - 1$ nach n gibt es nach Induktionsvoraussetzung $\mathcal{L}_3^+$-Formeln A_0, A_u und A_1 mit:

$$\begin{array}{rcl} f(A_0)(w_1, \ldots, w_{n-1}) & = & g(w_1, \ldots, w_{n-1}, 0) \\ f(A_u)(w_1, \ldots, w_{n-1}) & = & g(w_1, \ldots, w_{n-1}, u) \\ f(A_1)(w_1, \ldots, w_{n-1}) & = & g(w_1, \ldots, w_{n-1}, 1) \end{array}$$

Seien $x_1, \ldots, x_{n-1}$ die Aussagenvariablen in A_0, A_u, A_1 und x_n eine neue Aussagenvariable. Wir benötigen die folgenden aussagenlogischen Hilfsformeln:

$$\begin{array}{rcl} H_0 & = & \sim\sim \neg x_n \\ H_u & = & \sim x_n \wedge \sim \neg x_n \\ H_1 & = & \sim\sim x_n \end{array}$$

Setzt man jetzt

$$A = (A_0 \wedge H_0) \vee (A_u \wedge H_u) \vee (A_1 \wedge H_1),$$

so erhält man $f(A) = g$. Die Bedeutung der Hilfsformeln wird durch folgende Wahrheitstabelle ersichtlich.

Wahrheitstafel für die Hilfsformlen H			
x_n	H_1	H_u	H_0
1	1	0	0
u	0	1	0
0	0	0	1

■

Beispiel 6 *Gegeben sei die zweistellige Funktion g*

g	*1*	u	*0*
1	*0*	u	*0*
u	u	u	u
0	*0*	u	*0*

Man erhält leicht:

$$\begin{aligned} g(A,1) &= f(A \wedge \neg A) \\ g(A,u) &= u \\ g(A,0) &= f(A \wedge \neg A) \end{aligned}$$

Somit ergibt sich nach Konstruktion:

$$g(A,B) = f([A \wedge \neg A \wedge \sim\sim B] \vee [u \wedge \sim B \wedge \sim \neg B] \vee [A \wedge \neg A \wedge \sim\sim \neg B])$$

Das Konstruktionsverfahren liefert bei weitem nicht die kürzeste Formel, die eine gegebene Funktion realisiert. Es gilt im vorliegenden Beispiel

$$g(A,B) = f([A \wedge \neg A] \vee [B \wedge \neg B])$$

Beispiel 7 *Gegeben sei die zweistellige Funktion $\wedge_B$, die von Bochvar als Wahrheitstafel für die Konjunktion benutzt wurde.*

$\wedge_B$	*1*	u	*0*
1	*1*	u	*0*
u	u	u	u
0	*0*	u	*0*

Man erhält nach dem Konstruktionsverfahren:

$$\begin{aligned} A \wedge_B B \ id \ & (\neg A \wedge A \wedge \sim\sim \neg B) \vee \\ & (u \wedge \sim B \wedge \sim \neg B) \vee \\ & (A \wedge \sim\sim B) \end{aligned}$$

Vereinfachungen führen zu:

$$A \wedge_B B \ id \ (A \wedge B) \vee (A \wedge \neg A) \vee (B \wedge \neg B)$$

Tabelle aller einstelligen dreiwertigen Funktionen

X	$X\vee\sim X$	$\sim\neg X\vee u$	$\sim\neg X$	$X\vee\neg X$	$X\vee u$
1	1	1	1	1	1
u	1	1	1	u	u
0	1	u	0	1	u

X	X			$\sim\sim X$	$\sim X\vee u$
1	1	1	1	1	u
u	u	0	0	0	1
0	0	1	u	0	1

X			$u\vee\sim\sim\neg X$	u	$u\wedge X$
1	u	u	u	u	u
u	1	1	u	u	u
0	u	0	1	u	0

X			$u\wedge\sim\sim X$	$\sim X$	
1	u	u	u	0	0
u	0	0	0	1	1
0	1	u	0	1	u

X	$\sim X\wedge\sim\neg X$	$\neg X$	$\sim X\wedge u$	$X\wedge\neg X$	$\sim\sim\neg X$
1	0	0	0	0	0
u	1	u	u	u	0
0	0	1	u	0	1

X	$u\wedge\sim\sim\neg X$	$\sim\sim X\wedge\sim\neg X$			
1	0	0			
u	0	0			
0	u	0			

2.2.1 Die Postsche Logiken

Im zweiten Teil dieses Unterkapitels wollen wir funktionale Vollständigkeitssätze in einem allgemeineren Rahmen betrachten. Die erste Verallgemeinerung besteht darin, daß wir anstelle der dreiwertigen Logik $\mathcal{L}_3$ beliebige m-wertige Logiken betrachten. Wir beschränken uns hier auf Aussagenlogik. Die Erweiterungen auf die jeweilige Quantorenlogik sind leicht zu bewerkstelligen.

Als ein Beispiel betrachten wir die m-wertigen Logiken P_m von Emil Post, siehe [Post, 1921]. Die Menge der Wahrheitswerte von P_m sind die natürlichen Zahlen von 0 bis $m-1$, $W_m = \{0, \ldots, m-1\}$. Die Operatoren sind $\vee$ und der einstellige Operator s. Die Funktionen $\vee_P, s_P$ in der Matrix für P_m sind definiert durch

$$\begin{aligned} a \vee_P b &= \max(a, b) \\ s_P(a) &= a - 1 \,(\mathrm{mod}\ m) \end{aligned}$$

Hierbei sind $-$, max, mod m die Operationen auf den ganzen Zahlen $0, \ldots, m-1$ in ihrer üblichen Bedeutung. Schließlich sei $D = \{m-1\}$.

Wir wollen untersuchen, welche Funktionen in P_m definiert sind, d. h. sich in der Form $f(A)$ für eine aussagenlogische Formel A darstellen lassen und beginnen mit einem einfachen Lemma.

Lemma 5 *Die Funktionen*

$$x - k \,(\mathit{mod}\ m) \qquad \textit{für jedes } k, (0 \le k < m)$$

und

$$\mathit{max}\ (x_1, \ldots, x_r)$$

und

$$x + k \,(\mathit{mod}\ m) \qquad \textit{für jedes } k, (0 \le k < m)$$

sind in P_m definierbar.

Beweis: Es gilt offensichtlich:

$$x - k = \underbrace{s \ldots s}_{k \text{ mal}}(x)\,(\mathrm{mod}\ m)$$

$$\max\,(x_1, \ldots, x_r) = x_1 \vee_P \ldots \vee_P x_r$$

und

$$x + k = x - (m - k)\,(\mathrm{mod}\; m) \qquad \text{für jedes } k, (0 < k < m)$$

Für $k = 0$ setzt man natürlich $x + k = x$.

■

Wir schreiben im folgenden, wenn keine Mißverständnisse zu befürchten sind, vereinfachend $a - b, a + b$ anstelle von $a - b\,(\mathrm{mod}\; m), a + b\,(\mathrm{mod}\; m)$.

Wir beweisen zunächst ein einfaches, allgemeines Kriterium für die funktionale Vollständigkeit mehrwertiger Logiken, das nicht auf die Postschen Logiken beschränkt ist. Darin spielen die einstelligen Funktionen J_k für $0 \leq k < m$ eine herausragende Rolle:

$$J_k(x) = \begin{cases} m - 1 & \text{falls } x = k \\ 0 & \text{sonst} \end{cases}$$

Identifiziert man die Elemente $\{0, u, 1\}$ in der angegebenen Reihenfolge mit $\{0, 1, 2\}$, dann erhält man die folgende Entsprechung zwischen J_k und den im Beweis Lemma 4 von definierten Hilfsfunktionen:

$$J_0 = H_0, J_1 = H_u, J_2 = H_1$$

Lemma 6 (Einfaches Kriterium für funktionale Vollständigkeit)
Eine m-wertige Logik, in der die Funktionen

$$min(x, y), max(x, y), J_k \textit{ für } 0 \leq k < m$$

und alle konstanten Funktionen definierbar sind, ist vollständig.
Die Voraussetzung über die konstanten Funktionen besagt, daß für jedes $n > 1$ *und jeden Wahrheitswert* k, $0 \leq k < m$ *die Funktion* $c_k^n : \mathcal{M}^n \to \mathcal{M}$ *mit* $c_k^n(x_1, \ldots, x_n) = k$ *definierbar ist.*

Beweis:
Sei $g(\bar{x}) : \mathcal{M}^n \to \mathcal{M}$ gegeben.
$g(\bar{x}) = max\{min(c^n_{g(\bar{a})}, J_{a_1}(x_1), \ldots, J_{a_n}(x_n)) \mid \bar{a} = \langle a_1 \ldots a_n \rangle \in \mathcal{M}^n\}$

■

Wir kehren zurück zur Untersuchung definierbarer Funktionen in den Postschen Logiken P_m und zeigen als einen weiteren Vorbereitungsschritt zur funktionalen Vollständigkeit:

Lemma 7 *Für jedes m ist jede einstellige Funktion in P_m definierbar.*

Beweis:
Die Funktion

$$T(x) = \max(x, x-1, \ldots, x-m+1)$$

ist nach Lemma 5 definierbar in P_m und es gilt für alle $n \in \mathcal{M} = \{0, \ldots, m-1\}$:

$$T(n) = m - 1.$$

Der wesentliche Schritt des Beweises ist die Einführung der P_m- definierbaren Funktionen $T_k(x)$ für $k, 0 \leq k < m$:

$$T_k(x) = \max(\max[T(x) - 1, x] - m + 1, x + k) - m + 1$$

Für $x = m - 1$ berechnet sich $T_k(x)$ zu:

$$\begin{aligned} & \max\ (\ \max(m-2, m-1) - m + 1, m - 1 + k) - m + 1 \\ = \ & \max\ (0, m - 1 + k) - m + 1 \\ = \ & m - 1 + k - m + 1 = k \end{aligned}$$

Für $x < m - 1$ rechnet man aus:

$$\begin{aligned} & \max(\ \max(m-2, x) - m + 1, x + k) - m + 1 \\ = \ & \max(m - 2 - m + 1, x + k) - m + 1 \\ = \ & \max(m - 1, x + k) - m + 1 \\ = \ & m - 1 - m + 1 = 0 \end{aligned}$$

Man beachte, daß bei der zur dritten Zeile führenden Umformung $-1 = m - 1 (\mathrm{mod}\ m)$ benutzt wurde. Also insgesamt

$$T_k(x) = \begin{cases} 0 & \text{falls } x \neq m - 1 \\ k & \text{falls } x = m - 1 \end{cases}$$

Ist $g : \mathcal{M} \to \mathcal{M}$ eine beliebige einstellige Funktion so setzen wir $k_i = g(i), 0 \leq i < m$, und erhalten

$$g(x) = \max(T_{k_{m-1}}(x), T_{k_{m-2}}(x+1), \ldots, T_{k_0}(x + m - 1)).$$

oder kompakter geschrieben

$$g(x) = \max\{T_{k_{m-i-1}}(x + i) \mid 0 \leq i < m\}.$$

Zum Nachweis dieser Gleichung beachte man, daß für jedes $n \in \mathcal{M}$ und jedes i mit $0 \leq i < m$ gilt:

$$T_{k_{m-i-1}}(n+i) = 0,$$

falls $n + i \neq m - 1$ und

$$T_{k_{m-i-1}}(n+i) = k_{m-i-1} = k_n.$$

falls $n + i = m - 1$ Damit

$$\max(T_{k_{m-1}}(n), \ldots, T_{k_0}(n+m-1)) = k_n = g(n)$$

∎

Nach diesen Vorbereitungen können wir zeigen:

Theorem 8 *Die m-wertige Postsche Logik P_m ist funktional vollständig.*

Beweis:
Aus Lemma 7 folgt insbesondere die P_m-Definierbarkeit der Funktion $m - 1 - x$ (nicht zu verwechseln mit der Funktion $x - m$). Dann ist aber auch die Minimumsfunktion P_m-definierbar durch:

$$\min(x, y) = m - 1 - \max(m - 1 - x, m - 1 - y).$$

Nach Lemma 7 sind alle J_k P_m-definierbar. Ebenso sind alle einstelligen konstanten c_k^1 Funktionen P_m-definierbar. Wegen

$$c_k^n(x_1, \ldots, x_n) = c_k^1(min(x_1, \ldots, x_n))$$

sind die Voraussetzung des Kriteriums 6 erfüllt und damit P_m funktional vollständig.

∎

Die Untersuchung der funktionalen Vollständigkeit, insbesondere die Bereitstellung möglichst allgemeiner Kriterien ist ein stark entwickeltes Teilgebiet der mehrwertigen Logik. Eine gute Übersicht findet man in [Rosenberg, 1984]. Wir zitieren hier ohne Beweis ein typisches Resultat um wenigstens einen ungefähren Eindruck von der Richtung der Forschung auf diesem Gebiet zu geben. Siehe auch Aufgabe 2.2.7.

Satz 9 (Kriterium von Słupecki) *Sei $k > 2$ und H eine Menge von Funktionen auf der Menge $\{0, 1 \ldots k-1\}$ in sich selbst, die alle einstelligen Funktionen umfaßt.*

H ist funktional vollständig genau dann, wenn es eine Funktion $f \in H$ gibt, so daß

1. *f hängt nicht nur von einer Argumentstelle ab und*
2. *f ist surjektiv.*

Beweis: siehe [Słupecki, 1972].

Aus der zweiwertigen Aussagenlogik kennt man das Phänomen, daß alle aussagenlogischen Formeln mit einer einzigen Operation, dem Shefferstroke, aufgebaut werden können. Das gibt es auch in der m-wertigen Aussagenlogik. Wir betrachten dazu die Funktion

$$w(x, y) = \max(x, y) - 1 (\text{mod } m)$$

Offensichtlich gilt

$$w(x, x) = x - 1 = s_P(x)$$

und

$$\begin{aligned} s_P^{m-1}(w(x, y)) &= s_P(\ldots s_P(w(x, y)) \ldots) \\ &= \max(x, y) - m \\ &= \max(x, y) (\text{mod } m) \end{aligned}$$

Da sich die Basisfunktionen der Postschen Logik durch $w(x, y)$ darstellen lassen, läßt sich nach obigem Theorem jede Funktion von $\mathcal{M}$durch $w(x, y)$ darstellen.

2.2.2 Übungsaufgaben

Übungsaufgabe 2.2.1

Seien A, B aussagenlogische Ł3-Formeln. Dann gilt: Die Funktionen $f(A)$, $f(B)$ sind identisch, $f(A) = f(B)$, gdw. $(A \text{ id } B)$ ist eine Tautologie.

Übungsaufgabe 2.2.2 *Zeigen Sie die folgenden Tautologien:*

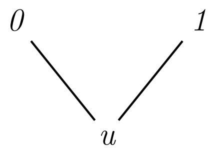
1. $(\neg\neg A \ id\ A)$
2. $(\neg \sim A \ id\ \sim\sim A)$
3. $(A \wedge B) \vee C \ id\ (A \vee C) \wedge (B \vee C)$
4. $(A \vee B) \wedge C \ id\ (A \wedge C) \vee (B \wedge C)$
5. $(A \vee B) \vee C \ id\ A \vee (B \vee C)$
6. $(A \wedge B) \wedge C \ id\ A \wedge (B \wedge C)$
7. $A \wedge B \ id\ B \wedge A$
8. $A \vee B \ id\ B \vee A$
9. $\neg(A \vee B) \ id\ \neg A \wedge \neg B$
10. $\neg(A \wedge B) \ id\ \neg A \vee \neg B$
11. $\neg\exists x A \ id\ \forall x \neg A$
12. $\neg\forall x A \ id\ \exists x \neg A$
13. $(\sim\sim\sim A \ id\ \sim A)$
14. $\sim (A \wedge B) \ id\ \sim A \vee \sim B$
15. $\sim (A \vee B) \ id\ \sim A \wedge \sim B$
16. $\sim \forall x A \ id\ \exists x \sim A$
17. $\sim \exists x A \ id\ \forall x \sim A$

Lösungshinweise: Machen Sie, wenn möglich, von der min bzw. max Definition Gebrauch, anstatt die Wahrheitstafeln auszurechnen.

Übungsaufgabe 2.2.3 *Finden Sie eine Ł3-Formel A, die folgende Funktion g realisiert:*

g	*1*	*u*	*0*
1	*1*	*u*	*u*
u	*u*	*u*	*u*
0	*u*	*u*	*0*

Bemerkung: Die Funktion g entsteht, indem man die drei Wahrheitswerte partiell ordnet als:



und $g(A,B) = \inf(A,B)$ setzt.

Übungsaufgabe 2.2.4 *Wegen der funktionalen Vollständigkeit der Postschen Logiken P_m sind die Funktionen $J_k(x)$ für $0 \leq k < m$ auch P_m-definierbar. Finden Sie durch Inspektion des Beweises von Lemma 7 eine Definition von J_k in Termini von T_j.*

Übungsaufgabe 2.2.5 *Wir wollen in dieser Aufgabe ein Kriterium für funktionale Vollständigkeit betrachten, das ein kleines bißchen stärker ist als das einfache Kriterium aus Lemma 6. Das Kriterium ist stärker, weil seine Voraussetzungen etwas schwächer sind. Wir benötigen zuerst die folgende*

Definition 18 (H-Abschluß) *Ist H eine Menge von Funktionen auf $\mathcal{M} = \{0,\ldots,m-1\}$, so heißt eine Teilmenge $N \subseteq \mathcal{M}$ H-**abgeschlossen**, falls für jede Funktion h in H, sagen wir h sei k-stellig, und für jedes k-Tupel $\bar{n}$ von Elementen aus N auch $h(\bar{n})$ in N liegt. Für eine beliebige Teilmenge $X \subseteq \mathcal{M}$ ist der H-**Abschluß** von X die kleinste H-abgeschlossene Menge, die X enthält.*

Die Menge H kann natürlich auch 0-stellige Funktionen enthalten, d.h. Konstanten. Aus der Definition folgt, daß eine H-abgeschlossene Menge jede in H vorkommende Konstante enthalten muß.
Der Begriff einer H-abgeschlossenen Menge ist die Verallgemeinerung des bekannten Unteralgebrabegriffs. Ist $\mathcal{G} = (G, e, \cdot, inv)$ eine Gruppe mit

Trägermenge G, neutralem Element e, Gruppenoperation $\cdot$ und der Inversenfunktionen inv, dann sind für $H = \{e, \cdot, inv\}$ die H-abgeschlossenen Teilmengen von G genau die Untergruppen von $\mathcal{G}$ und der H-Abschluß einer Teilmenge $X \subseteq G$ entspricht der von X erzeugten Untergruppe von $\mathcal{G}$. Entsprechend läßt sich der Begriff eines Unterraumes von Vektorräumen für eine geeignete Wahl von H als Spezialfall H-abgeschlossener Mengen erkennen.
Zeigen Sie das folgende Kriterium für funktionale Vollständigkeit

Sei H eine Menge von Funktionen auf $W_m = \{0, \ldots, m-1\}$, die max, min und alle $J_k, 0 \leq k < m$ enthält. Dann ist eine n-stellige Funktion g H-definierbar

gdw.

für jedes n-Tupel $a_1, \ldots, a_n$ von Elementen aus W_m liegt $g(a_1, \ldots, a_n)$ im H-Abschluß von $\{a_1, \ldots, a_n\}$.

Übungsaufgabe 2.2.6

Definition 19 1. *Eine Funktion $f : W_m^2 \to W_m$ heißt* **multiplikativ**, *falls für alle $x \in W_m$ gilt*

$$f(0, x) = f(x, 0) = 0 \text{ und } f(m-1, x) = f(x, m-1) = x$$

2. *Eine Funktion $g : W_m^2 \to W_m$ heißt* **additiv**, *falls für alle $x \in W_m$ gilt*

$$g(0, x) = g(x, 0) = x$$

Sei f eine beliebige multiplikative und g eine beliebige additive Funktion auf W_m. Eine Menge H von Funktionen, die f,g, J_k für $0 \leq k < m$ und alle konstanten Funktionen enthält ist funktional vollständig.
Entnommen aus [Muzio & Wesselkamper, 1986] Theorem 2.3 auf Seite 4.

Übungsaufgabe 2.2.7 *Sei $\mathcal{M}$eine endliche Algebra. Falls für $k \leq 2$ jede k-stellige Funktion in $\mathcal{M}$definierbar ist, dann ist auch jede $(k+1)$-stellige Funktion definierbar.*
Dieses Resultat wurde zuerst in [Post, 1921] bewiesen.

A ⇒ B			
B ↓ A →	1	u	0
1	1	1	1
u	u	1	1
0	0	u	1

Abbildung 2.12: Wahrheitstafeln für ⇒

Übungsaufgabe 2.2.8 *Die Wahrheitstabelle in Abbildung 2.12 definiert die Implikation ⇒ von Łukasiewics. Sie unterscheidet sich im Fall* $u \Rightarrow 0$ *von der schwachen Implikation und im Fall* $u \Rightarrow u$ *von der starken Implikation. Berechnen Sie eine Formel in* $\mathcal{L}_3^+$*, die ⇒ definiert.*

Übungsaufgabe 2.2.9 *Zeigen Sie, daß das Kriterium von Słupecki aus Satz 9 für* $k = 2$ *nicht gilt.*

2.3 Kalküle

In diesem Kapitel wenden wir uns der Frage zu, wie man in mehrwertigen Logiken rechnen kann. Die historisch gesehen ersten Antworten auf diese Frage waren vollständige und korrekte Axiomensysteme im Hilbertschen Stil. Wir zitieren ein Resultat als kleine Kostprobe. Weitere Axiomatisierungsergebnisse findet man in der angegebenen Literatur, vor allem in [Bolc & Borowik, 1992].

Satz 10 *Sei $\mathcal{L}^3_{Łukas}$ die Aussagenlogik mit den Konnektoren $\neg$ (starke Negation) und $\Rightarrow$ (Łukasiewicz Implikation, siehe Abb. 2.12). Zusammen mit der modus ponens Regel bilden die folgenden Axiome ein vollständiges und korrektes Axiomensystem für $\mathcal{L}^3_{Łukas}$:*

1. $(A \Rightarrow (B \Rightarrow A))$
2. $(A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow C))$
3. $(\neg A \Rightarrow \neg B) \Rightarrow (B \Rightarrow A)$
4. $((A \Rightarrow \neg A) \Rightarrow A) \Rightarrow A$

Beweis: Das Resultat wurde zuerst in [Wajsberg, 1931] bewiesen. Heute kann man diese Axiomatisierung als einen Spezialfall eines allgemeinen Theorems von Rosser und Turquette erhalten, siehe z.B. [Malinowski, 1993], Kapitel 8. Eine Verallgemeinerung dieser Axiomatisierung auf die Prädikatenlogik ist leicht möglich, durch die Hinzunahme der aus der zweiwertigen Logik bekannten Axiomen und Regeln für Quantoren. Näheres findet man z.B. in [Gottwald, 1990], Seite 56.

2.3.1 Allgemeine Tableaukalküle

Wir wollen ausführlich auf Tableaukalküle für mehrwertige Logiken eingehen. In den Arbeiten [Surma, 1984], [Carnielli, 1987] und [Hähnle, 1993] werden Verfahren vorgestellt, die zu jeder endlich-wertigen Logik einen vollständigen und korrekten Tableaukalkül herstellen. Legt man eine Logik $\mathcal{L}$ fest durch die Wahl einer endlichen Menge M von Wahrheitswerten, einer Menge D

von ausgezeichneten Wahrheitswerten und einer Liste von Wahrheitstabellen für beliebig gewählte aussagenlogische Operatoren F, dann liefern diese Verfahren einen für $\mathcal{L}$ vollständigen und korrekten Tableaukalkül $TK_{\mathcal{L}}$, wobei in [Carnielli, 1987] auch noch verallgemeinerte Quantoren für $\mathcal{L}$ betrachtet werden können. Die Zuordnung von $TK_{\mathcal{L}}$ zu $\mathcal{L}$ ist außerdem recht einfach und kann leicht automatisiert werden.
Wir werden im folgenden zunächst den allgemeinen Rahmen für die in [Hähnle, 1993] betrachteten Tableaukalküle vorstellen und anschließend als ein konkretes Beispiel einen Kalkül für $\mathcal{L}_3$ im Detail vorstellen. Wir wollen Verallgemeinerungen der Quantoren aus unseren Betrachtungen ausklammern. Die einzigen Quantoren in allen mehrwertigen Logiken $\mathcal{L}$, die wir betrachten werden, sind also $\forall$ und $\exists$.
Wie schon in dem Standardwerk [Smullyan, 1968] für den zweiwertigen Tableaukalkül sollen auch hier signierte Formeln oder, wie wir sagen, Vorzeichenformeln benutzt werden. Hier wie dort sind signierte Formeln ein technisches Hilfsmittel, auf das im Prinzip verzichtet werden kann. Im mehrwertigen Fall scheint aber mit der Einführung signierter Formeln neben der notationellen Bequemlichkeit auch eine Steigerung der Effizienz eines tableaubasierten Beweisers verbunden zu sein.

Definition 20
Ist $\mathcal{L}$ eine beliebige mehrwertige Logik mit M als Menge der Wahrheitswerte, so ist eine Vorzeichenformel für $\mathcal{L}$ von der Form

$$SA$$

wobei A eine Formel in $\mathcal{L}$ ist und S eine nichtleere Teilmenge von M.

Eine Vorzeichenformel ohne freie Variablen SA ist in einer $\mathcal{L}_3$-Struktur $(\mathcal{M}, v)$ gültig, wenn der Wahrheitswert von A bzgl. $(\mathcal{M}, v)$ in S liegt, in Zeichen:

$$(\mathcal{M}, v) \models SA, \text{ genau dann, wenn } v(A) \in S.$$

Zu jedem Vorzeichen $\emptyset \neq S \subseteq M$ und jedem aussagenlogischen Operator $\circ$ wird es eine Tableauregel geben. Sei $\circ$ ein n-stelliger Operator, für ein beliebiges $n \in \mathbb{N}$. Die zugehörige Tableauregel hat dann die Form

$$\frac{S \circ (A_1, \ldots, A_n)}{T(A_1, \ldots, A_n)}$$

$$
\begin{array}{c|c|c|c|c}
\multicolumn{5}{c}{\{4\}(p \supset q)} \\
\hline
\{0\}p & \{0,1\}p & \{0,1,2\}p & \{0,1,2,3\}p & \\
 & \{1,2,3,4\}q & \{2,3,4\}q & \{3,4\}q & \{4\}q
\end{array}
$$

Abbildung 2.13: Tableauregel aus der 5-wertigen Łukasiewicz Logik

wobei $T(A_1, \ldots, A_n)$ ein endliches erweitertes Tableau ist, in dem nur Vorzeichenformeln der Form $S'A_i$ auftreten. Ein erweitertes Tableau, kann im Unterschied zu den bisherigen Tableaux mehrere Wurzeln $W_1, \ldots, W_k$ besitzen. Die Teiltableaux T_i von $T(A_1, \ldots, A_n)$ mit den Wurzeln W_i nennt man die **Extensionen** der Regel. Ein Beispiel einer solchen Regel ist in Abbildung 2.13 zu sehen. Diese Regel besitzt die fünf Extensionen:

$$
\begin{array}{l}
\{\{0\}p\} \\
\{\{0,1\}p, \{1,2,3,4\}q\} \\
\{\{0,1,2\}p, \{2,3,4\}q\} \\
\{\{0,1,2,3\}p, \{3,4\}q\} \\
\{\{4\}q\}
\end{array}
$$

Tableaux sind mit den Tableauregeln erzeugte endlich Bäume[1], deren Knoten signierte Formeln sind. Ein Pfad in einem Baum, der mit der Wurzel beginnt und in einem Blatt endet, heißt **Zweig**. Meist identifizieren wir einfach einen Zweig mit der Menge der signierten Formeln, die seine Knotenmenge ist. Jede Tableauregel hat eine signierte Formel als **Prämisse**. Die **Konklusion** einer Regel ist, wie schon erwähnt, eine Menge von Extensionen. Die einzelnen Extensionen stellen dabei Alternativen dar, wie ein Modell für die Formel in der Prämisse gebildet werden kann. Wie wir in den folgenden Beweisen sehen werden, stehen die Regeln in direkter Beziehung zu den Modellbedingungen. Nach diesen einleitenden Erklärungen kommen wir jetzt zu den formalen Definitionen.

Definition 21 (Tableau)
Ein **Tableau** *für eine endliche Menge Σ signierter Formeln wird folgendermaßen konstruiert:*

[1]Was ein Baum sei, setzen wir als bekannt voraus. Eine völlig formale Definition der hier verwendeten Bäume findet man in [Smullyan, 1968].

1. *Ein linearer Baum, in dem jede Formel aus Σ genau einmal vorkommt, ist ein Tableau für Σ.*

2. *Sei T ein Tableau für Σ und B ein Zweig in T, der eine signierte Formel SA enthält. Weiterhin gebe es eine Tableauregel R mit Prämisse SA. Sind $E_1, \ldots, E_n$ die Extensionen von R (unter Berücksichtigung der genannten Einschränkungen, falls R eine Quantorenregel ist), so wird T am Ende von B um n lineare Unterbäume erweitert, die jeweils die signierten Formeln aus den E_i in beliebiger Reihenfolge enthalten. Der so entstehende Baum ist wieder ein Tableau für Σ.*

Definition 22
Ein Zweig B eines Tableau ist **geschlossen**, *falls gilt:*

1. *B enthält signierte Formeln $S_1A_1, \ldots S_kA_k$, so daß eine Substitution σ alle A_i unifiziert, d.h. $\sigma(A_1) = \ldots = \sigma(A_k)$ und*

$$\bigcap_{i=1}^{i=k} S_i = \emptyset$$

2. *B enthält eine signierte Formel SA, auf die keine Regel anwendbar ist und A ist nicht atomar.*

Ein Zweig, der nicht geschlossen ist, heißt **offen**.

Bemerkung:

- Der erste Punkt dieser Definition ist eine Verallgemeinerung der Existenz eines komplementären Paares. Dieser Fall ist für $k = 2$ in der obigen Definition mitenthalten. Ein Zweig, der für eine Formel A die signierten Formeln $\{0, 1\}A$, $\{1, 2\}A$ und $\{2, 0\}A$ enthält ist sicherlich nicht erfüllbar, obwohl er kein komplementäres Paar enthält.

Definition 23

1. *Ein Tableau heißt* **geschlossen**, *falls jeder Zweig darin geschlossen ist, sonst heißt es* **offen**.

2. *Ein Tableau T heißt ein* **systematisches** *Tableau, wenn auf jedem offenen Zweig B von T*

   (a) *für jede Vorzeichenformel SC auf B, wobei C nicht atomar ist, die entsprechende Tableauregel angewandt wird und*

   (b) *außerdem für jede Formel der Form $T\forall xA$, $U\forall xA$, $F\exists xA$, $U\exists xA$ oder $\{U, F\}\exists xA$ (solche Formeln nennen wir γ-**Formeln**), die auf B vorkommt, unendlich oft die zugehörige Regel anwendet wird.*

   Einen Zweig B, der diese beiden Bedingungen erfüllt, nennen wir einen **erschöpften** *Zweig.*

Bemerkungen:

- In beiden Fällen der Definition 23 eines geschlossenen Zweigs wurde ein Widerspruch erzeugt, was bedeutet, daß der entsprechende Zweig nicht zu einem Modell vervollständigt werden kann.
- Im aussagenlogischen Fall ist es klar, daß jedes systematische Tableau für eine Formelmenge Σ endlich sein muß, wenn Σ endlich ist. Daher ist jede Tableaukonstruktion, die systematisch ein geschlossenes Tableau zu erstellen versucht, eine Entscheidungsprozedur für die Aussagenlogik. Im prädikatenlogischen Falle dagegen können erschöpfte Zweige unendlich viele Formeln enthalten, so daß ein systematisches Tableau nur dann garantiert endlich ist, wenn ein geschlossenes Tableau existiert.

Definition 24
Sei T ein Tableau, in dem die freien Variablen $\bar{y} = \langle y_1, \ldots\ldots, y_k\rangle$ auftreten. T heißt **erfüllbar***, wenn es eine $\mathcal{L}_3$-Struktur $(\mathcal{M}, v)$ gibt, so daß für jedes k-Tupel $\bar{a} = \langle a_1, \ldots, a_k\rangle$ von Elementen aus $\mathcal{M}$ mindestens ein Pfad P in T existiert, so daß für alle signierten Formeln B auf P*

$$(\mathcal{M}, v) \models B(\bar{a}/\bar{y})$$

gilt.

Definition 25
Eine Menge aussagenlogischer Tableauregeln für eine mehrwertige Logik $\mathcal{L}$ heißt

- **korrekt** *wenn für jede Regel mit der Prämisse* $SA \circ B$ *und den Extensionen* $E_1, \ldots, E_k$ *mit*

  $$E_i = \{S_{ij}C_{ij} \mid 1 \leq j \leq k_i \leq 2\} \text{ wobei } C_{ij} \in \{A, B\}$$

  und für jede $\mathcal{L}_3$*-Struktur* $(\mathcal{M}, v)$*mit*

  $$v_{\mathcal{M}}(A \circ B) \in S$$

  ein i, $1 \leq i \leq k$ *existiert, so daß für alle* j *gilt*

  $$v_{\mathcal{M}}(C_{ij}) \in S_{ij}$$

- **vollständig** *wenn für jede Regel mit der Prämisse* $SA \circ B$ *und den Extensionen* $E_1, \ldots, E_k$ *mit*

  $$E_i = \{S_{ij}C_{ij} \mid 1 \leq j \leq k_i \leq 2\} \text{ wobei } C_{ij} \in \{A, B\}$$

  und für jede $\mathcal{L}_3$*-Struktur* $(\mathcal{M}, v)$*, welche für ein* i*,* $1 \leq i \leq k$ *und für alle* j *gilt*

  $$v(C_{ij}) \in S_{ij}$$

  erfüllt, auch

  $$v(A \circ B) \in S$$

  gilt.

  Für jedes zusammengesetzte Vorzeichenformel $SA \circ B$*, die nicht als Prämisse einer Tableauregel vorkommt gilt für jede* $\mathcal{L}_3$*-Struktur* $(\mathcal{M}, v)$

  $$v(A \circ B) \notin S.$$

Es genügt dabei A *und* B *als aussagenlogische Variablen zu betrachten.*

Entsprechende Forderungen an Regeln für die Quantoren sind schwieriger zu formulieren. Man bräuchte dazu ein Äquivalent zu den Wahrheitstafeln einer aussagenlogischen Verknüpfung. Es ist zwar möglich ein solches Konzept zu finden, siehe etwa [Carnielli, 1987, Carnielli, 1991], aber die Komplexität dieses Begriffs steht in keinem Verhältnis zu seinen bisherigen Anwendungsmöglichkeiten. Wir betrachten daher im allgemeinen Fall nur die aussagenlogischen Verknüpfungen und Quantoren nur in dem Beispiel im nächsten Abschnitt.

Satz 11
Sei T *ein erfüllbares Tableau.* T_1 *entstehe aus* T *durch Anwendung einer korrekten aussagenlogischen Tableauregel, dann ist auch* T_1 *erfüllbar.*

Beweis: Sei $(\mathcal{M}, v)$ eine erfüllende $\mathcal{L}_3$-Struktur für T. Wir zeigen, daß $(\mathcal{M}, v)$ auch T_1 erfüllt. Sei also $\bar{a} = \langle a_1, \ldots, a_k \rangle$ ein beliebiges k-Tupel von Elementen aus $\mathcal{M}$. Wir müssen zeigen, daß es einen Pfad P_1 in $T_!$ gibt, so daß für alle signierten Formeln B auf P_1 gilt $(\mathcal{M}, v) \models B(\bar{a}/\bar{y})$. Nach Voraussetzung gibt es einen Pfad P in T, so daß $(\mathcal{M}, v) \models B(\bar{a}/\bar{y})$ für alle B auf T gilt. Wurde beim Übergang von T zu T_1 der Pfad P nicht erweitert, dann sind wir fertig. Betrachten wir den kritischen Fall, daß P durch eine Regelanwendung erweitert wurde. Es gibt also eine signierte Formel $SA = SA_1 \circ A_2$ auf P und eine Tableauregel mit Prämisse SA und Extensionen $E_1, \ldots E_k$ und P wurde erweitert zu den Pfaden $P_i' = P \cup E_i$ in T_1. Nach Voraussetzung gilt $v(A_1 \circ A_2) \in S$. Da die Regel als korrekt angenommen wurde gibt es eine Extension E_i, $1 \leq i \leq k$, so daß für alle signierten Formeln $S_{ij}C_{ij} \in E_i$ gilt $v_{\mathcal{M}}(C_{ij}) \in S_{ij}$. Damit ist ein erfüllender Pfad $P_1 = P \cup E_i$ in T_1 gefunden.

■

Korollar 12

Sei A eine $\mathcal{L}$-Formel für eine mehrwertige Logik $\mathcal{L}$. Sei $S = W \setminus D$, wobei W die Menge aller Wahrheitswerte und D die Menge der ausgezeichneten Wahrheitswerte von $\mathcal{L}$ ist. Falls ein geschlossenes Tableau T mit Wurzelmarkierung SA existiert, dann ist A eine $\mathcal{L}$-Tautologie.

Beweis: Wir führen einen Widerspruchsbeweis. Wenn A keine Tautologie ist, dann ist SA erfüllbar. Folglich ist auch des Tableau T_0, das nur aus der mit SA markierten Wurzel besteht erfüllbar. Durch wiederholte Anwendung von Satz 11 folgt die Erfüllbarkeit von T. Das steht aber im Widerspruch zur Definition eines geschlossenen Tableau.

■

Satz 13

Sei A eine aussagenlogische $\mathcal{L}$-Tautologie. Dann gibt es ein geschlossenes Tableau für SA.
Hierbei bezeichnet S dieselbe Menge wie in 12.

Beweis: ■

2.3.2 Ein Tableaukalkül für $\mathcal{L}_3$

Als **Vorzeichen** für $\mathcal{L}_3$-Formeln treten ein- oder zweielementige Teilmengen oder Wahrheitswerte T, U, F auf, das heißt die Mengen

$$\{F\}, \{T\}, \{U\}, \{F, T\}, \{F, U\}, \{T, U\}.$$

In dem betrachteten Tableaukalkül werden nur die Vorzeichen $\{T\}, \{U\}, \{F\}$ und $\{U, F\}$ auftreten. In den Tableauregeln für $\mathcal{L}_3$enthält die komplizierteste Regel (das ist die Regel für $\{U\}A \wedge B$) eine Konklusion mit drei Extensionen. Alle auftretenden Extensionen bestehen aus höchstens zwei signierten Formeln.

Die Regeln des Tableaukalküls

Regeln für $\wedge$:

$$\frac{\{T\}A \wedge B}{\begin{array}{c}\{T\}A\\ \{T\}B\end{array}} \qquad \frac{\{U\}A \wedge B}{\begin{array}{ccccc}\{U\}A & & \{U\}B & & \{U\}A\\ \{T\}B & \text{oder} & \{T\}A & \text{oder} & \{U\}B\end{array}}$$

$$\frac{\{F\}A \wedge B}{\{F\}A \quad \text{oder} \quad \{F\}B} \qquad \frac{\{U, F\}A \wedge B}{\{U, F\}A \quad \text{oder} \quad \{U, F\}B}$$

Regeln für $\vee$:

$$\frac{\{T\}A \vee B}{\{T\}A \quad \text{oder} \quad \{T\}B} \qquad \frac{\{U\}A \vee B}{\begin{array}{ccc}\{U, F\}A & & \{U\}A\\ \{U\}B & \text{oder} & \{F\}B\end{array}}$$

$$\frac{\{F\}A \vee B}{\begin{array}{c}\{F\}A\\ \{F\}B\end{array}} \qquad \frac{\{U, F\}A \vee B}{\begin{array}{c}\{U, F\}A\\ \{U, F\}B\end{array}}$$

Regeln für $\sim$:

$$\frac{\{T\} \sim A}{\{U,F\}A} \qquad \frac{\{F\} \sim A}{\{T\}A} \qquad \frac{\{U,F\} \sim A}{\{T\}A}$$

Regeln für $\neg$:

$$\frac{\{T\}\neg A}{\{F\}A} \qquad \frac{\{F\}\neg A}{\{T\}A} \qquad \frac{\{U\}\neg A}{\{U\}A} \qquad \frac{\{U,F\}\neg A}{\{T\}A \quad \text{oder} \quad \{U\}A}$$

Regeln für $\nabla = \sim \neg$:

$$\frac{\{T\}\nabla A}{\{T\}A \quad \text{oder} \quad \{U\}A} \qquad \frac{\{F\}\nabla A}{\{F\}A} \qquad \frac{\{U,F\}\nabla A}{\{F\}A}$$

Regeln für $\supset$:

$$\frac{\{T\}A \supset B}{\{U,F\}A \quad \text{oder} \quad \{T\}B} \qquad \frac{\{F\}A \supset B}{\begin{array}{c}\{T\}A\\ \{F\}B\end{array}} \qquad \frac{\{U\}A \supset B}{\begin{array}{c}\{T\}A\\ \{U\}B\end{array}} \qquad \frac{\{U,F\}A \supset B}{\begin{array}{c}\{T\}A\\ \{U,F\}B\end{array}}$$

Regeln für $\forall$:

$$\frac{\{T\}\forall x A(x)}{\{T\}A(z)} \qquad \frac{\{F\}\forall x A(x)}{\{F\}A(f(y_1,\ldots,y_k))}$$

$$\frac{\{U\}\forall x A(x)}{\begin{array}{c}\{U\}A(f(y_1,\ldots,y_k))\\ \{T\}\nabla A(z)\end{array}} \qquad \frac{\{U,F\}\forall x A(x)}{\{U,F\}A(f(y_1,\ldots,y_k))}$$

Hierbei steht ∇ für den einstelligen aussagenlogischen Operator $\sim \neg$.

Regeln für $\exists$:

$$\frac{\{T\}\exists x A(x)}{\{T\}A(f(y_1,\ldots,y_k))} \qquad \frac{\{F\}\exists x A(x)}{\{F\}A(z)}$$

$$\frac{\{U\}\exists x A(x)}{\begin{array}{c}\{U\}A(f(y_1,\ldots,y_k))\\ \{T\}\sim A(z)\end{array}} \qquad \frac{\{U,F\}\exists x A(x)}{\{U,F\}A(z)}$$

$$\frac{\{T\}\forall x A(x)}{\{T\}A(z)} \qquad \frac{\{F\}\forall x A(x)}{\{F\}A(f(y_1,\ldots,y_k))}$$

$$\frac{\{U\}\forall x A(x)}{\begin{array}{c}\{U\}A(f(y_1,\ldots,y_k))\\ \{T\}A(z) \quad oder \quad \{U\}A(z)\end{array}} \qquad \frac{\{U,F\}\forall x A(x)}{\{U,F\}A(f(y_1,\ldots,y_k))}$$

$$\frac{\{T\}\exists x A(x)}{\{T\}A(f(y_1,\ldots,y_k))} \qquad \frac{\{F\}\exists x A(x)}{\{F\}A(z)}$$

$$\frac{\{U\}\exists x A(x)}{\begin{array}{c}\{U\}A(f(y_1,\ldots,y_k))\\ \{T\}\sim A(z)\end{array}} \qquad \frac{\{U,F\}\exists x A(x)}{\{U,F\}A(z)}$$

Abbildung 2.14: Tableauregeln für Quantoren in $\mathcal{L}_3$

Zusätzlich zur Beschreibung der Quantorenregeln sind für die Anwendung dieser Regeln die folgenden Einschränkungen wichtig:

- z ist eine neue Variable, d. h. eine Variable, die zum Zeitpunkt der Regelanwendung noch nicht als freie Variable im Tableau vorkommt.
- $y_1, \ldots, y_k$ sind genau die freien Variablen in $\forall x A(x)$
- f ist ein neues k-stelliges Funktionszeichen (Skolemfunktionszeichen). Im Falle $k = 0$ ist f eine neue Skolemkonstante.

Die Konstruktion eines Tableau für eine Formelmenge kann als Versuch aufgefaßt werden, auf systematische Weise ein Modell für die Formelmenge zu finden. Kommt dieser Konstruktionsprozeß an einen Punkt von dem aus ersichtbar ist, daß keine erfolgreiche Fortsetzung mehr möglich ist - das dann erreichte Tableau nennt man ein **geschlossenes** Tableau - so kann die Formelmenge kein Modell besitzen, ist also nicht erfüllbar. Besteht die Formelmenge aus nur einer Formel, heißt dies im zweiwertigen Fall, das Negat der

Formel ist gültig. Bei mehrwertigen Logiken ist "nicht falsch" nicht mehr identisch mit "wahr". Die änderung des Wahrheitswerts einer Formel kann nicht durch einfache Negation ausgedrückt werden, weswegen es sinnvoll ist, mit **Vorzeichenformeln** im Tableau zu arbeiten.

Lemma 14
Alle aussagenlogischen Regeln für $\mathcal{L}_3$ sind korrekt und vollständig (siehe Definition 25).

Beweis: Wir zeigen exemplarisch die Korrektheit der Regel für $\{U,F\}(A \land B)$. Die Formel $\{U,F\}(A \land B)$ liege auf dem Pfad P in T. T_1 entsteht aus T, indem P zum einen zu $P_1 = \langle P, \{U,F\}A\rangle$ und zum anderen zu $P_2 = \langle P, \{U,F\}B\rangle$ verlängert wird. Sei $\bar{a}$ ein k-Tupel aus $\mathcal{M}$. Nach Voraussetzung gibt es einen Pfad P' in T, so daß für alle C auf P'

$$(\mathcal{M}, v) \models C(\bar{a}/\bar{y})$$

gilt. Ist $P' \neq P$, so bleibt diese Aussage auch für T_1 richtig, und wir sind fertig. Der interessante Fall ist somit $P' = P$. Da insbesondere $(\mathcal{M}, v) \models \{U,F\}(A \land B)(\bar{a}/\bar{y})$ gilt, folgt, wie eine Inspektion der Wahrheitstabelle für $\land$ zeigt,

$$(\mathcal{M}, v) \models \{U,F\}A(\bar{a}/\bar{y})$$

oder

$$(\mathcal{M}, v) \models \{U,F\}B(\bar{a}/\bar{y})$$

D. h. es gilt

$$(\mathcal{M}, v) \models P_1(\bar{a}/\bar{y})$$

oder

$$(\mathcal{M}, v) \models P_2(\bar{a}/\bar{y})$$

Insgesamt ist die Erfüllbarkeit von T_1 gezeigt.

Die Verifikation aller übrigen aussagenlogischer Regeln verläuft nach dem gleichen Schema und sei dem Leser überlassen.

∎

Lemma 15 *Sei T ein endliches erfüllbares Tableau und T_1 entstehe aus T durch* **eine** *Regelanwendung, dann ist auch T_1 erfüllbar.*

Beweis: Entsteht T_1 aus T durch Anwendung einer aussagenlogischen Regel, so folgt der Beweis aus Lemma 14 und Satz refsatz:korrektschrittallgem. Ist $(\mathcal{M}, v)$ eine $\mathcal{L}_3$-Struktur, die T erfüllt, dann soll gezeigt werden, daß $(\mathcal{M}, v)$, nachdem eventuell neue Skolemfunktionen geeignet interpretiert werden, auch T_1 erfüllt. Wir betrachten exemplarisch ein Beispiel mit Quantoren. Wir nehmen an, die Formel $U\forall xA(x)$ komme auf P in T vor. T_1 entsteht, indem P zu $P_1 = \langle P, UA(f(y_1, \ldots, y_k)), T\nabla A(z)\rangle$ verlängert wird.

Ist $(\mathcal{M}, v)$ eine $\mathcal{L}_3$-Struktur, die T erfüllt, dann soll gezeigt werden, daß $(\mathcal{M}, v)$, nachdem eventuell neue Skolemfunktionen geeignet interpretiert werden, auch T_1 erfüllt.

Als erstes wollen wir die Skolemfunktion f in $(\mathcal{M}, v)$ interpretieren.

Falls ein b existiert mit $v_{\mathcal{M}}(A(b/x, \bar{a}/\bar{y}) = u$, so fixieren wir ein solches b und setzen

$$f^{\mathcal{M}}(\bar{a}) = b.$$

Andernfalls wählen wir einen beliebigen Funktionswert

$$f^{\mathcal{M}}(\bar{a}) = c.$$

Zur notationellen Vereinfachung bezeichnen wir auch die neue $\mathcal{L}_3$-Struktur noch mit $(\mathcal{M}, v)$.

Wir wollen zeigen, daß T_1 in $(\mathcal{M}, v)$ erfüllbar ist. Sei dazu ein Elementtupel $\langle \bar{a}, d\rangle$ fixiert. In T_1 ist die neue freie Variable z hinzugekommen, deswegen heben wir das zur Belegung von z dienende Element d des Tupels eigens hervor. Der einzig interessante Fall liegt vor, wenn für alle Formeln C auf P

$$(\mathcal{M}, v) \models C(\bar{a}/\bar{y})$$

gilt. Insbesondere gilt dann

$$v(\forall xA(x, \bar{a}/\bar{y})) = u.$$

Es gibt somit mindestens ein b mit

$$v(A(b/x, \bar{a}/\bar{y})) = u$$

und deswegen nach Definition f:

$$v(A(f(\bar{y})/x)(\bar{a}/\bar{y})) = u.$$

Außerdem gilt für alle d

$$v(A(d/x, \bar{a}/\bar{y})) \neq 0,$$

was gleichbedeutend ist zu

$$v(\nabla A(d/x, \bar{a}/\bar{y})) = 1.$$

Zusammenfassend haben wir

$$(\mathcal{M}, v) \models P_1(d/z, \bar{a}/\bar{y})$$

gezeigt. Die Verifikation aller übrigen Quantorenregeln verläuft nach dem gleichen Schema und sei dem Leser überlassen.

■

Satz 16 (Korrektheitssatz)
Sei A eine $\mathcal{L}_3$-Formel ohne freie Variablen. Falls es ein Tableau T mit Wurzel $\{U, F\}A$ gibt, in dem jeder Pfad geschlossen ist, dann ist A eine $\mathcal{L}_3$-Tautologie.

Beweis: Wäre A keine $\mathcal{L}_3$-Tautologie, dann wäre $\{U, F\}A$ erfüllbar. Durch wiederholte Anwendung von Lemma 15 wüßten wir, daß dann auch T erfüllbar ist. Das steht im Widerspruch zur Voraussetzung, daß alle Pfade in T geschlossen und damit nicht erfüllbar sind.

■

Korollar 17 *Sei Σ eine Menge von $\mathcal{L}_3$-Formeln ohne freie Variablen und A eine weitere Formel dieser Art. Falls es ein Tableau T über Σ mit Wurzel $\{U, F\}A$ gibt, im dem jeder Pfad geschlossen ist, dann gilt*

$$\Sigma \vdash_3 A.$$

Beweis: Gilt $\Sigma \vdash_3 A$ nicht, dann wäre $\Sigma \cup \{\{U, F\}A\}$ erfüllbar. Wie im Beweis von Satz 16 ergibt sich daraus ein Widerspruch.

■

Satz 18 Vollständigkeitssatz
Sei A eine $\mathcal{L}_3$-Tautologie. Dann ist jedes systematische Tableau mit Wurzel $\{U, F\}A$ geschlossen.

Beweis: Angenommen die Aussage des Satzes sei falsch. Dann gibt es ein systematisches, erschöpftes Tableau mit Wurzel $\{U,F\}A$, das einen offenen Zweig P enthält. Sei H_0 die Menge aller signierten Formeln, die auf P vorkommen. Insbesondere liegt $\{U,F\}A$ in H_0. Die Formeln in H_0 können freie Variable enthalten, die bei der Anwendung von Quantorenregeln eingeführt werden. Da wir von einem systematischen Tableau ausgegangen sind, kommt für jede Formel $B(z)$ in H_0 mit einer freien Variablen z, die Formel $B(z_i)$ für unendlich viele verschiedene Variablen z_i ebenfalls in H_0 vor. Die Formelmenge H entstehe aus H_0, indem jede freie Variable in H_0 durch einen variablenfreien Term ersetzt wird. Dabei wird natürlich die gleiche Variable in allen Formeln in H_0 durch denselben Term ersetzt. Da es nur abzählbar unendlich viele Terme gibt, kann der Ersetzungsprozeß außerdem so eingerichtet werden, daß für jede Formel $B(z^1,\ldots,z^k)$ in H_0, für jedes k-Tupel variablenfreier Terme $t^1,\ldots,t^k$ die Formel $B(t^1,\ldots,t^k)$ in H liegt. Weiter ist zu beachten, daß die in P neu eingeführten Skolemfunktionen und -konstanten beim Aufbau der variablenfreien Terme mitbenutzt werden.

Die Formelmenge H hat die folgenden Eigenschaften

Konsistenz:
Für atomare, variablenfreie Formeln p und disjunkte Vorzeichen S_1 und S_2 kommen S_1p und S_2p nicht beide in H vor.

$\wedge$:
Liegt $T(A \wedge B)$ in H, dann auch $TA \in H$ und $TB \in H$.
Liegt $F(A \wedge B)$ in H, dann $FA \in H$ oder $FB \in H$.
Liegt $U(A \wedge B)$ in H, dann $(UA \in H$ und $TB \in H)$
oder $(UB \in H$ und $TA \in H)$
oder $(UA \in H$ und $UB \in H)$.
Liegt $\{U,F\}(A \wedge B)$ in H, dann $\{U,F\}A \in H$ oder $\{U,F\}B \in H$.

Diese $\wedge$-Eigenschaft für H folgt, ebenso wie die folgenden, aus der Tatsache, daß wir ein systematisches Tableau vorausgesetzt hatten, und aus der Form der Tableauregeln.

$\vee$:
Liegt $T(A \vee B)$ in H, dann $TA \in H$ oder $TB \in H$.
Liegt $F(A \vee B)$ in H, dann $FA \in H$ und $FB \in H$.
Liegt $U(A \vee B)$ in H, dann $(\{U,F\}A \in H$ und $UB \in H)$

oder ($UA \in H$ und $FB \in H$) .

Liegt $\{U,F\}(A \vee B)$ in H, dann $\{U,F\}A \in H$ und $\{U,F\}B \in H$.

$\sim$:

Liegt $T \sim A$ in H, dann $\{U,F\}A \in H$.
Liegt $F \sim A$ in H, dann $TA \in H$.
Liegt $\{U,F\} \sim A$ in H, dann $TA \in H$.

Die signierte Formel $U \sim A$ liegt nicht in H. Für sie existiert keine Tableauregel. Ihr Auftreten würde P zu einem geschlossenen Zweig machen.

$\neg$:

Liegt $T\neg A$ in H, dann $FA \in H$.
Liegt $F\neg A$ in H, dann $TA \in H$.
Liegt $U\neg A$ in H, dann $UA \in H$.
Liegt $\{U,F\}\neg A$ in H, dann $TA \in H$ oder $UA \in H$.

∇:

Liegt $T\nabla A$ in H, dann $TA \in H$ oder $UA \in H$.
Liegt $F\nabla A$ in H, dann $FA \in H$.
Liegt $\{U,F\}\nabla A$ in H, dann $FA \in H$.

$U\nabla A$ liegt nicht in H.

$\supset$:

Liegt $T(A \supset B)$ in H, dann $\{U,F\}A \in H$ oder $TB \in H$.
Liegt $F(A \supset B)$ in H, dann $TA \in H$ und $FB \in H$.
Liegt $U(A \supset B)$ in H, dann $TA \in H$ und $UB \in H$.
Liegt $\{U,F\}(A \supset B)$ in H, dann $TA \in H$ und $\{U,F\}B \in H$.

$\forall$:

Liegt $T\forall xA(x)$ in H, dann liegt für jeden variablenfreien Term t $TA(t/x)$ in H.
Liegt $F\forall xA(x)$ in H, dann gibt es einen variablenfreien Term t, so daß $FA(t/x)$ in H liegt.
Liegt $U\forall xA(x)$ in H, dann gibt es einen variablenfreien Term t, so daß $UA(t/x)$ in H liegt, und für jeden variablenfreien Term s liegt $TA(s/x)$ in H oder $UA(s/x)$.
Liegt $\{U,F\}\forall xA(x)$ in H, dann gibt es einen variablenfreien Term t, so daß $\{U,F\}A(t/x) \in H$.

Zum Nachweis des zweiten Teils der $\forall$-Eigenschaft von H beachten wir zunächst, daß die Formel $F\forall xA(x)$ durch Ersetzung der freien Variablen

$y_1, \ldots, y_k$ durch Terme $t_1, \ldots, t_k$ aus einer Formel $F\forall x A(x, y_1, \ldots, y_k)$ in H_0 entsteht. Durch Anwendung der entsprechenden $\forall$-Regel ist dann auch $FA(f(y_1, \ldots, y_k)/x)$ in H_0, und wir können den gewünschten Term t als $t = f(t_1, \ldots, t_k)$ erhalten.

Zum Nachweis des dritten Teils beachte man, daß wir außer von der entsprechenden $\forall$-Regel auch von der ∇-Eigenschaft von H Gebrauch gemacht haben.

$\exists$:
Liegt $T\exists x A(x)$ in H, dann gibt es einen variablenfreien Term t, so daß $TA(t/x) \in H$.
Liegt $F\exists x A(x)$ in H, dann liegt für jeden variablenfreien Term t $FA(t/x)$ in H.
Liegt $U\exists x A(x)$ in H, dann gibt es einen variablenfreien Term t mit $UA(t/x) \in H$, und für alle Terme s gilt $\{U, F\}A(s/x) \in H$.
Liegt $\{U, F\}\exists x A(x)$ in H, dann liegt für jeden variablenfreien Term t die Formel $\{U, F\}A(t/x)$ in H.

Nach der Aufzählung der notwendigen Eigenschaften der Formelmenge H können wir nun den Beweis von Satz 18 zu Ende bringen. Wir erinnern uns daran, daß wir begonnen haben, einen Widerspruchsbeweis zu führen. Wir werden den gesuchten Widerspruch erhalten, indem wir eine erfüllende $\mathcal{L}_3$-Struktur für $\{U, F\}A$ finden. Wir werden sogar eine $\mathcal{L}_3$-Struktur $(\mathcal{M}, v)$ angeben, die jede Formel aus H wahr macht. Das Universum von $\mathcal{M}$bestehe aus allen variablenfreien Termen, die Interpretation der Funktions- und Konstantenzeichen sei wie in Herbrand-Strukturen. Es bleibt für jede variablenfreie, atomare Formel p der Funktionswert $v(p)$ zu erklären:

$$\begin{array}{ll} \text{Gilt} & Tp \in H, \text{ dann } v(p) = 1. \\ \text{Gilt} & Fp \in H, \text{ dann } v(p) = 0. \\ \text{Gilt} & Up \in H, \text{ dann } v(p) = u. \end{array}$$

Liegt keiner dieser drei Fälle vor und $\{U, F\}p \in H$, dann $v(p) = 0$.
Kommt überhaupt keine signierte Formel Sp in H vor, dann setzen wir willkürlich $v(p) = 0$.

Wegen der Konsistenzeigenschaft ist die Definition von $v(p)$ eindeutig.

Man zeigt jetzt durch Induktion über den Formelaufbau von A, daß jede Formel $SA \in H$ in $(\mathcal{M}, v)$ wahr ist. Diese Verifikation ist recht umfangreich, wir begnügen uns mit einem typischen Beispiel.

Es liege $U(A \wedge B)$ in H. Wegen der $\wedge$-Eigenschaft von H gilt einer der drei Fälle:

$$\begin{array}{lcl} UA \in H & \text{und} & TB \in H. \\ UB \in H & \text{und} & TA \in H. \\ UA \in H & \text{und} & UB \in H. \end{array}$$

Nehmen wir an, der dritte Fall liegt vor. Da A und B von geringerer Komplexität sind als $(A \wedge B)$, sind nach Induktionsvoraussetzung UA und UB in $(\mathcal{M}, v)$ wahr, d. h. $v(A) = v(B) = u$. Damit gilt aber auch $v(A \wedge B) = u$ nach der $\wedge$-Wahrheitstabelle. Die Argumentation in den ersten beiden Fällen verläuft völlig analog.

■

Beispiel 8 *Sei t ein Term, dessen einzige Variable y ist. Dann ist*

0. $C = (\forall x Q(x)) \supset \forall y Q(t/x)$

eine Tautologie.

1. $\{U, F\}C$ *(aus 0)*

2. $T\forall x Q(x)$ *(aus 1)*

3. $\{U, F\}\forall y Q(t/x)$ *(aus 1)*

4. $TQ(z_1)$ *(aus 2)*

5. $\{U, F\}Q(s/x)$ *(aus 3)*

wobei $s = t(c/y)$. Der einzige Zweig dieses Tableaus kann durch $z_1 \rightarrow s$ geschlossen werden (die Zeilen 4 und 5 enthalten dann komplementäre Formeln).

Wir betrachten ein etwas umfangreicheres Beispiel, um die Notwendigkeit der wiederholten Anwendung von Regeln für γ-Formeln zu illustrieren.

Beispiel 9 *Sei* Σ *die folgende Formelmenge*

$\forall x \forall y\ (nachbar(x, y) \supset\ weg(x, y))$
$\forall x \forall y \forall z\ (nachbar(x, z) \wedge\ weg(z, y) \supset\ weg(x, y))$
$nachbar(a, b)$
$nachbar(b, c)$
$nachbar(c, d)$

Offensichtlich gilt $\Sigma \vdash$ weg(a, d). Wir wollen ein geschlossenes Tableau über Σ mit Wurzelmarkierung $\{U, F\}$ weg(a, d) konstruieren. In den folgenden Diagrammen wird der mit Formeln aus Σ markierte Teil nicht mitgezeichnet.

Tab:

$$1.\quad \{U,F\}\ \text{weg}(a,d)$$

$$2.(\Sigma)\quad T\ \text{nachbar}(x,z) \wedge\ \text{weg}(z,y) \supset\ \text{weg}(x,y)$$

$3.(2)$ T weg(x,y) $4.(2)$ $\{U,F\}$nachbar$(x,z) \wedge$ weg(z,y)

closed$(1,3)$

$x \to a$

$y \to d$ $5.(4)$ $\{U,F\}$nachbar(x,z) $6.(4)$ $\{U,F\}$weg(z,y)

closed$(5,\Sigma)$

$x \to a$ Tab$_1$

$z \to b$

Tab$_1$:

$$6.\quad \{U,F\}\text{weg}(b,d)$$

$$7.(\Sigma)\quad T\ \text{nachbar}(u,v) \wedge weg(v,w) \supset\ \text{weg}(u,w)$$

$8.(7)$ T weg(u,w) $9.(7)$ $\{U,F\}$nachbar$(u,v) \wedge$ weg(u,w)

closed$(6,8)$

$u \to b$

$w \to d$10.(9) $\{U,F\}$nachbar(u,v) $11.(9)$ $\{U,F\}$weg(v,w)

closed$(10,\Sigma)$

$u \to b$ Tab$_2$

$v \to c$

Tab$_2$:

$$11.(9)\quad \{U,F\}\text{weg}(c,d)$$

$$12.(\Sigma)\quad T\ \text{nachbar}(z_1,z_2) \supset\ \text{weg}(z_1,z_2)$$

$13.(12)$ T weg(z_1,z_2) $14.(12)$ $\{U,F\}$nachbar(z_1,z_2)

closed$(11,13)$ closed$(14,\Sigma)$

$z_1 \to c$ $z_1 \to c$

$z_2 \to d$ $z_2 \to d$

Ein weiteres Beispiel ist in Abbildung 2.15 zu sehen.

Basisalgorithmus

Eingabe: Ein nicht geschlossener Zweig T

Ausgabe: **true**, falls der Zweig geschlossen werden kann, **fail** sonst.

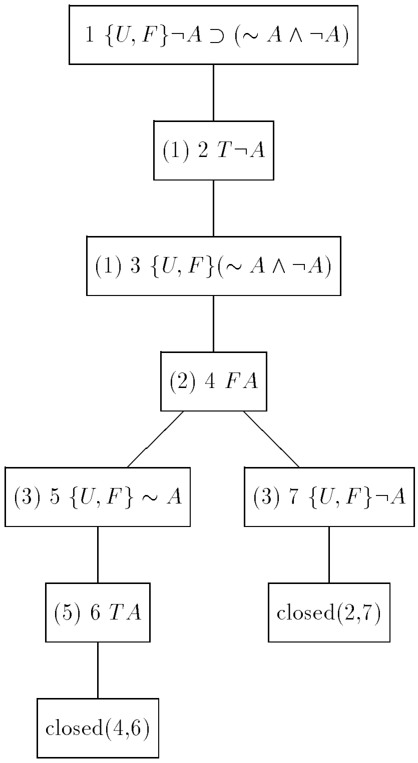


Abbildung 2.15: Beispiel eines Tableaubeweises in der Logik $\mathcal{L}_3$

1. {Priorisierung der Formeln} Zunächst werden die Formeln priorisiert in bezug auf gewisse Heuristiken. [Es ist zu beachten, daß bei verzweigenden Regeln die Anzahl effektiver Nachfolger (das heißt Zweige, die nicht sofort geschlossen werden können,) berechnet wird. Zweige, die dabei geschlossen werden, sollen später nicht noch einmal betrachtet werden müssen. Die Information muß also (inklusive möglicher Instantiierungen) gerettet werden.] Das Ergebnis dieses Teilabschnitts ist eine Formel P mit höchster Priorität (†) und eine zugehörige Regel R_P.

2. {Regelanwendung} Wende R_P auf P an und erzeuge einen oder mehrere neue Zweige.
   Danach wird P entweder aus der Liste potentieller Kandidaten für eine Regelanwendung entfernt oder der Zähler BOUND wird erhöht.
   In diesem Schritt werden auch die Nachfolgeranzahlen der zu den neu

generierten Formeln korrespondierenden Regeln ermittelt, und es erfolgt eine entsprechende Eintragung in die Repräsentation des Zweigs.

3. {Direktes Schließen möglichst vieler Zweige, Finden erschöpfter Zweige} Für jeden neuen Zweig:

   (a) Versuche, den Zweig zu schließen
   d. h. Formeln ohne korrespondierende Regel oder Unifikation (†) von zwei komplementären Formeln im aktuellen Zweig; hier hat man einen Freiheitsgrad, indem man beliebige Formeln testet, oder auch nur Atome).

   (b) Falls ein Zweig erschöpft ist, fertig mit Ausgabe **fail**.

4. {Schließen der noch offenen Zweige} Für alle Zweige, die noch offen sind:

   (a) Rufe Basisalgorithmus rekursiv auf.

5. Fertig mit Ausgabe **true**.

Im Algorithmus sind die möglichen Backtracking-Punkte mit (†) gekennzeichnet.

In dem einführenden Kapitel zur Logik $\mathcal{L}_3$hatten wir in Lemma 3 auf Seite 30 bewiesen, daß eine Formel A, die nur mit $\wedge, \vee \sim, \forall, \exists$ aufgebaut ist, genau dann eine dreiwertige Tautologie ist, wenn sie eine zweiwertige Tautologie ist. Der Beweis wurde damals modelltheoretisch geführt. Wir können jetzt einen zweiten Beweis dieses Lemmas führen, indem wir sagen:

Es gibt ein geschlossenes zweiwertiges Tableau für FA
gdw.
es gibt ein geschlossenes dreiwertiges Tableau für $\{U, F\}A$.

Wir müssen zunächst erklären, was die Regeln für das zweiwertige Tableauverfahren sind. Das sind die angegebenen Regeln für die Vorzeichen T und F, wobei allein die Regel

$\dfrac{\{T\} \sim B}{\{U, F\}B}$ ersetzt werden muß durch

A *cand* B			
B ↓ A →	1	u	0
1	1	u	0
u	u	u	0
0	0	u	0

A *cor* B			
B ↓ A →	1	u	0
1	1	u	1
u	1	u	u
0	1	u	0

Abbildung 2.16: Wahrheitstabellen für *cand* und *cor*

$$\frac{\{T\} \sim B}{\{F\}B}$$

(siehe z. B. [Smullyan 68])

Durch Inspektion der Regel verifiziert man die folgenden beiden Behauptungen:

1. Ist T_2 ein geschlossenes zweiwertiges Tableau für FA, so erhält man ein geschlossenes dreiwertiges Tableau, wenn man überall in T_2 das Vorzeichen F durch das Vorzeichen $\{U, F\}$ ersetzt.

2. Ist T_3 ein geschlossenes dreiwertiges Tableau für $\{U, F\}A$, so erhält man ein geschlossenes zweiwertiges Tableau für FA, wenn man überall in T_3 das Vorzeichen $\{U, F\}$ durch das Vorzeichen F ersetzt.

2.3.3 Übungsaufgaben

Übungsaufgabe 2.3.1 *In der Tableauregel für $U\forall$ hätte man den neuen aussagenlogischen Operator ∇ vermeiden können und dafür das bisher nicht benutzte Vorzeichen $\{T, U\}$ verwenden können, also*

$$\frac{\{U\}\forall x A(x)}{\begin{array}{c}\{U\}A(f(y_1, \ldots, y_k))\\ \{T, U\}A(z)\end{array}}$$

Dadurch hätte man sich auch alle Tableauregeln für ∇ gespart. Welche zusätzlichen Regeln hätte man aber gebraucht?

Übungsaufgabe 2.3.2 *Welche Tableauregeln müßte man für cand, cor, siehe Abb. 2.16 hinzunehmen, um einen korrekten und vollständigen Tableaukalkül zu erhalten?*

Bewerkenswert ist die Asymmetrie der beiden Konnektoren, die eine dreiwertige Version der Konjunktion bzw. Disjunktion sein sollen. Die den Konnektoren cand und cor zugrunde liegende Vorstellung geht davon aus, daß die Berechnung des Wahrheitswertes eine Formel entweder terminiert und den Wert 0 *oder* 1 *liefert oder nicht terminiert. Im letzteren Fall wird der Wahrheitswert u vergeben. Liefert die Berechnung des Wahrheitswertes von A den Wert* 0*, so wird auch der Wahrheitswert von A cand B gleich* 0 *gesetzt. Die Berechnung des Wahrheitswertes von B wird garnicht erst gestartet.*

Diese Junktoren werden in dem Buch [Gries, 1989] S. 69 definiert und sind dort Bestandteil einer dreiwertigen Logik zur Programmverifikation.

Übungsaufgabe 2.3.3 *Die aussagenlogische Verknüpfung* $\circ$ *sei durch die folgende Wahrheitstabelle bestimmt.*

$\circ$	1	u	0
1	1	1	1
u	1	1	1
0	1	0	1

Geben Sie die Tableauregeln für die Vorzeichenmengen $\{1\}, \{0\}$ *und* $\{0, 1\}$ *an.*
Diese Wahrheitstafel ist insofern interessant, da die Tautologien unter den aussagenlogischen Formeln, die ausschließlich mit $\circ$ *aufgebaut sind, mit* 1 *als dem einzigen ausgezeichneten Wahrheitswert nicht durch ein endliches Hilbertsches Axiomensystem axiomatisiert werden können. Siehe [Urquhart, 1986], Seite 85, wo auch weitere Literaturhinweise zu finden sind.*

Übungsaufgabe 2.3.4
Wir sagen, daß $\mathcal{M}$ *die Struktur* $\mathcal{N}$ *subsumiert, wenn* $\mathcal{M} \sqsubseteq^F \mathcal{N}$ *(siehe Definition 16 auf Seite 36) gilt für den Spezialfall, daß F die Identitätsabbildung ist. In diesem Fall schreiben wir einfach* $\mathcal{M} \sqsubseteq \mathcal{N}$

Definition 26

1. *Eine Formel* ϕ *der Logik* $\mathcal{L}_3$*heißt* **1-persistent***, wenn für je zwei* $\mathcal{L}_3$*-Strukturen* $\mathcal{M} \sqsubseteq \mathcal{N}$ *mit* $v_{\mathcal{M}}(\phi) = 1$ *auch* $v_{\mathcal{N}}(\phi) = 1$ *gilt*

2. *ϕ heißt* **0-persistent**, *wenn für je zwei $\mathcal{L}_3$-Strukturen $\mathcal{M} \sqsubseteq \mathcal{N}$ mit $v_{\mathcal{M}}(\phi) = 0$ auch $v_{\mathcal{N}}(\phi) = 0$ gilt*

3. *ϕ heißt* **persistent**, *wenn ϕ 1-persitent und 0-persistent ist.*

Sei c ein n-stelligee aussagenlogischer Junktor und $P_1, \ldots, P_n$ aussagenlogische Variablen. Dann ist $c(P_1, \ldots, P_n)$ 1-persistent genau dann, wenn es eine korrekte und vollständige Tableauregel

$$\frac{\{1\}c(P_1, \ldots, P_n)}{E_1 \mid \ldots \mid \quad E_k}$$

gibt, so daß in $E_1, \ldots, E_k$ nur die Vorzeichen $\{1\}$ und $\{0\}$ auftreten.

2.4 Anwendungen

Drei- und mehrwertige Logiken haben in folgenden Bereichen Anwendungen gefunden.

- Als technisches Hilfmittel zum Nachweis der Unabhängigkeit der Axiome in einem gegebenen Axiomensystem,
- zur Modellierung undefinierter Funktions- und Prädikatswerte in der Spezifikation und Verifikation von Programmen,
- in der Semantik natürlicher Sprache, z.B. zur Modellierung von Präsuppositionen, siehe dazu [Max, 1990],
- in der Theorie der logischen Programmierung zur Beschreibung des Zusammenhangs zwischen der operationalen Semantik der Negation und einem deklarativen Gegenstück,
- zur Modellierung von elektronischen Schaltkreisen. Die zusaätzlichen Wahrheitswerte werden entweder benutzt um physikalisch existierende Strom- bzw. Spannungszustände zu modellieren oder sie dienen dazu verschiedene Fehlerzustände und ihre Propagierung zu beschreiben.
- zur Modellierung von Vagheit und Unbestimmtheit in vielen Bereichen der Mathematik und Informatik, z.B. in der Theorie der Intervallarithmetik.
- zur Modellierung und Behandlung von abstrahierenden Prädikaten, wie z.B.in [Sagiv *et al.*, 1999].

2.4.1 Unabhängigkeitsbeweise

Eine der ersten Anwendungen mehrwertiger Logiken waren Unabhängigkeitsbeweise in logischen Axiomensystemen. Als Pionierarbeit ist hier vor allem [Bernays, 1926] zu nennen. Die folgende Darstellung orientiert sich an [Gottwald, 1990], Seite 62 ff. Um diese Technik zu erläutern, betrachten wir das in Tabelle 2.1 gezeigte Axiomensystem K_1 für die Aussagenlogik: Zusammen

Ax1 $p_1 \Rightarrow (p_2 \Rightarrow p_1)$

Ax2 $((p_1 \Rightarrow p_2) \Rightarrow p_1) \Rightarrow p_1$

Ax3 $(p_1 \Rightarrow p_2) \Rightarrow ((p_2 \Rightarrow p_3) \Rightarrow (p_1 \Rightarrow p_3))$

Ax4 $(p_1 \wedge p_2) \Rightarrow p_1$

Ax5 $(p_1 \wedge p_2) \Rightarrow p_2$

Ax6 $(p_1 \Rightarrow p_2) \Rightarrow ((p_1 \Rightarrow p_3) \Rightarrow p_1 \Rightarrow p_2 \wedge p_3))$

Ax7 $p_1 \Rightarrow (p_1 \vee p_2)$

Ax8 $p_2 \Rightarrow (p_1 \vee p_2)$

Ax9 $(p_1 \Rightarrow p_3) \Rightarrow ((p_2 \Rightarrow p_3) \Rightarrow p_1 \vee p_2 \Rightarrow p_3))$

Ax10 $(p_1 \approx p_2) \Rightarrow (p_1 \Rightarrow p_2)$

Ax11 $(p_1 \approx p_2) \Rightarrow (p_2 \Rightarrow p_1)$

Ax12 $(p_1 \Rightarrow p_2) \Rightarrow ((p_2 \Rightarrow p_1) \Rightarrow p_1 \approx p_2))$

Ax13 $(p_1 \Rightarrow p_2) \Rightarrow (\neg p_2 \Rightarrow \neg p_1)$

Ax14 $p_1 \Rightarrow \neg\neg p_1$

Ax15 $\neg\neg p_1 \Rightarrow p_1$

Tabelle 2.1: Das Axiomensystem K_1

$\Rightarrow$	1	u	0	$\approx$	1	u	0	p	$\neg p$
1	1	u	0	1	1	u	0	1	0
u	1	1	u	u	u	1	u	u	u
0	1	1	1	0	0	u	1	0	1

Tabelle 2.2: Wahrheitstabellen Łukasiewicz Junktoren

mit der modus ponens Regel

$$\frac{H, H \Rightarrow G}{G}$$

ist K_1 ein vollständiges Axiomensystem für die klassische Aussagenlogik, siehe z.B. [Asser, 1959]. Außerdem sind die Axiome von K_1 unabhängig voneinander, d.h. es kann keines weggelassen werden ohne die Vollständigkeit zu zerstören. Diese Eigenschaft läßt sich auch formulieren als:

Definition 27 *Ein Axiomensystem* $\mathbf{K}$ *heißt* unabhängig, *wenn für jedes Axiom* $A \in \mathbf{K}$, *in der Ableitungsrelation,* $\vdash_{\mathbf{K}\setminus\{A\}}$, *die durch dieselben Regeln und die um* A *verminderte Axiomenmenge von* $\mathbf{K}$ *gegeben ist, die Formel* A *nicht mehr herleitbar ist, d.h.*

$$\not\vdash_{\mathbf{K}\setminus\{A\}} \ A.$$

Wir wollen exemplarisch zeigen, daß Ax2 unabhängig ist von den restlichen Axiomen in $\mathbf{K}_1$. Dazu betrachten wir die dreiwertige Logik $L_{\mathbf{K}_1}$ mit den Wahrheitswerten $W = \{1, u, 0\}$, dem ausgezeichneten Wahrheitswert 1 und den Junktoren

$$\neg, \Rightarrow, \wedge, \vee, \approx,$$

die durch die Wahrheitstabellen von Łukasiewicz interpretiert werden, (siehe Tabelle 2.2 und 2.3).

Wir müssen drei Dinge nachweisen:

1. jedes Axiom in $\mathbf{K}_1$ außer Ax2 ist eine $L_{\mathbf{K}_1}$-Tautologie
2. alle Regeln des Kalküls (im vorliegenden Fall also nur die modus ponens Regel) führen von $L_{\mathbf{K}_1}$-Tautologien wieder zu einer $L_{\mathbf{K}_1}$-Tautologie

$\wedge$	1	u	0	$\vee$	1	u	0
1	1	u	0	1	1	1	1
u	u	u	0	u	1	u	u
0	0	0	0	0	1	u	0

Tabelle 2.3: Wahrheitstabellen Łukasiewicz Junktoren

3. Ax2 ist keine $L_{\mathbf{K}_1}$-Tautologie

Aus diesen drei Behauptungen folgt, daß jede Formel B, die aus $\mathbf{K}_1 \setminus \{Ax2\}$ abgeleitet werden kann, eine $L_{\mathbf{K}_1}$-Tautologie ist. Daher muß

$$\mathbf{K}_1 \setminus \{Ax2\} \not\vdash Ax2$$

gelten.

zu 1:
(Ax1):
Angenommen es gäbe eine Belegung v der Aussagenlogischen Variablen p_1, p_2, so daß $v(Ax1)$ kein ausgezeichneter Wahrheitswert ist, d.h. $v(Ax1) \in \{u, 0\}$. Die Wahrheitstabelle für $\Rightarrow$ zeigt, daß es hierfür zwei Möglichkeiten gibt:

$$\begin{array}{rcl}
v(p_1) = u \text{ und } v(p_2 \Rightarrow p_1) = 0 & \text{oder} & v(p_1) = 1 \text{ und } v(p_2 \Rightarrow p_1) \in \{u, 0\} \\
\\
\text{aus } v(p_2 \Rightarrow p_1) = 0 \; folgt & & \text{aus } v(p_2 \Rightarrow p_1) \in \{u, 0\} \; folgt \\
v(p_1) = 0 \text{ und } v(p_2) = 1 & & v(p_1) \in \{u, 0\} \\
\text{Widerspruch} & & \text{Widerspruch}
\end{array}$$

(Ax3):
Wie schon beim Nachweis von Axiom 1 nehmen wir an, daß es eine Bewertung v gäbe mit $v(Ax3) \in \{u, 0\}$ und führen diese Annahme durch erschöpfende Fallunterscheidung zum Widerspruch.
Fall 1:
1.$v(p_1 \Rightarrow p_2) = u$ und
2.$v((p_2 \Rightarrow p_3) \Rightarrow (p_1 \Rightarrow p_3)) = 0$
aus 2. folgt:
5.$v((p_2 \Rightarrow p_3) = 1$ und

$6. v(p_1 \Rightarrow p_3) = 0$
aus 6. folgt weiter:
$7. v(p_1) = 1$ und
$8. v(p_3) = 0$
aus 7. und 1. folgt:
$9. v(p_2) = u$
aus 9. und 8. folgt $v(p_2 \Rightarrow p_3) = u$ im Widerspruch zu 5.

Fall 2:
$3. v(p_1 \Rightarrow p_2) = 1$ und
$4. v((p_2 \Rightarrow p_3) \Rightarrow (p_1 \Rightarrow p_3)) \in \{u, 0\}$
Zeile 4 führt zu einer weiteren Fallunterscheidung:

Fall 2.1:
$10. v(p_2 \Rightarrow p_3) = u$ und
$11. v(p_1 \Rightarrow p_3) = 0$
aus 11. folgt:
$12. v(p_1) = 1$ und
$13. v(p_3) = 0$
aus 12. und 3. folgt jetzt $v(p_2) = 1$, während aus 13. und 10. $v(p_2) = u$ folgt.
Widerspruch !

Fall 2.2:
$14. v(p_2 \Rightarrow p_3) = 1$ und
$15. v(p_1 \Rightarrow p_3) =\in \{u, 0\}$
Die letzte Zeile führt zu einer weiteren Fallunterscheidung
Fall 2.2.1:
$16. v(p_1) = u$ und
$17. v(p_3) == 0$
aus 16. und 3. folgt $v(p_2) \in \{1, u\}$, während aus 17. und 14. $v(p_2) = 0$ folgt, ein Widerspruch.
Fall 2.2.2:
$18. v(p_1) = 1$ und
$19. v(p_3) =\in \{u, 0\}$
aus 18. und 3. folgt diesmal $v(p_2) = 1$ und 19. liefert mit 14. den Widerspruch $v(p_2) \in \{u, 0\}$

(Ax4):
Aus $v(p_1 \wedge p_2) \Rightarrow p_1) \in \{u, 0\}$ folgt entweder

$$v(p_1 \wedge p_2) = u \text{ und } v(p_1) = 0$$
$$\text{oder}$$
$$v(p_1 \wedge p_2) = 1 \text{ und } v(p_1) \in \{u, 0\}$$

In beiden Fällen ergibt sich ein Widerspruch zu der Eigenschaft der Konjunktion : $v(p_1 \wedge p_2) \leq v(p_1)$

(Ax5):
Analog zu (Ax4)

zu 2:
Ist $v(H) = 1$, so folgt aus der Wahrheitstafel für $\Rightarrow$, daß $v(H \Rightarrow G) = 1$ nur gelten kann, wenn auch $v(G) = 1$.

zu 3:

p_1	p_2	$p_1 \Rightarrow p_2$	$(p_1 \Rightarrow p_2) \Rightarrow p_1$	$((p_1 \Rightarrow p_2) \Rightarrow p_1) \Rightarrow p_1$
1	1	1	1	1
1	u	u	1	1
1	0	0	1	1
u	1	1	u	1
u	u	1	u	1
u	0	u	1	u
0	1	1	0	1
0	u	1	0	1
0	0	1	0	1

■

Die Anwendung dieses Verfahrens verlangt ein hohes Maß an Kreativität, denn für jeden Anwendungsfall muß eine geeignete mehrwertige Logik gefunden werden.

2.4.2 Logik mit partiellen Funktionen

VDM (Vienna Development Method) ist eine Methode zur formalen Spezifikation und Entwicklung von Computerprogrammen. Ein Bestandteil dieser

Methode ist die formale Spezifikationssprache VDM-SL. Als Referenz verweisen wir auf die Bücher [Jones, 1990] und [Bicarregui *et al.*, 1994]. Für uns ist interessant, daß zur Beschreibung der Semantik von VMD-SL eine dreiwertige Semantik benutzt wird, explizit in Abschnitt 3.3 von [Jones, 1990] und implizit in [Bicarregui *et al.*, 1994].

Partielle Funktionen sind ein Problem, das in jeder formalen Spezifikationssprache auftritt. Was soll ein Ausdruck der Form $\frac{x}{y}$ für reele Zahlen x, y bedeuten, wenn $y = 0$ ist, oder was soll die Differenz $n - m$ für natürliche Zahlen n, m bedeuten, wenn $m > n$ ist? Häufig wird dieses Problem unter den Teppich gekehrt oder mit der kurzen Bemerkung abgetan, daß durch die Einführung eines zusätzlichen Elements in das betrachtete Universum, das Problem einfach zu lösen sei. In VDM wurde eine gründlichere Lösung angestrebt. Zunächst sieht alles noch, wie erwartet aus: jedes Sortenuniversum wird um ein Element erweitert, das als Wert für funktionen dienen soll, für die sonst kein Wert existiert. Zum Universum $I\!N$der natürlichen Zahlen wird das Element $*_{I\!N}$ hinzugenommen. Damit lassen sich die folgenden Definitionen formal hinschreiben:

$$[\tfrac{n}{m}] = \begin{cases} \text{das kleinste } k \in I\!N \text{ mit } k \times m \geq n & \text{falls } m \neq 0 \\ *_{I\!N} & \text{sonst} \end{cases}$$

$$[n - m] = \begin{cases} n - m & \text{falls } n \geq m \\ *_{I\!N} & \text{sonst} \end{cases}$$

Bevor wir das weitere Vorgehen in VDM schildern betrachten wir ein Beispiel:

$$\forall n, m([\frac{n}{m}] = [m - n] \to 5n = 5m^2 - 5nm)$$

Betrachten wir $*_{I\!N}$ als neues Element des Universums der natürlichen Zahlen und lassen alles weitere unverändert, so ist diese Formel falsch; für $m = 0$ und $n = 1$ ist die Prämisse der Implikation wahr, da ja $[\frac{1}{0}] = *_{I\!N} = [0 - 1]$. Aber die rechte Seite der Implikation führt zu der Ungleichung $5 \neq 0$. Die Option für jeden nichtdefinierten Funktionswert ein neues Element einzuführen ist nicht sehr verlockend. Alternativ dazu, und das ist der Weg der in VDM beschritten wird, kann man die Gleichheitrelation ändern, wie in Abbbildung 2.17 gezeigt. Dabei ist $*_{I\!B}$ ein neues Element, das zu der Menge $I\!B$ der Boolschen Werte hinzugefügt wird, mit anderen Worten ein neuer Wahrheitswert. Die Wahrheitstafeln der dreiwertigen VDM-Logik stimmen in der in Abbil-

$A =_3 B$	$B \in \mathbb{N}$	$B \equiv *_{\mathbb{N}}$
$A \in \mathbb{N}$	$A = B$	$*_{\mathbb{B}}$
$A \equiv *_{\mathbb{N}}$	$*_{\mathbb{B}}$	$*_{\mathbb{B}}$

Abbildung 2.17: Dreiwertige Gleichheit

dung 2.18 angegbenen Weise mit den uns bekannten Wahrheitstafeln überein.

Korrespondenz der VDM-Logik mit $\mathcal{L}_3$	
VDM (nach Jones90)	$\mathcal{L}_3$
$true$	1
$*$	u
$false$	0
$\wedge, \vee$	$\wedge, \vee$ min, max
$\neg$	$\neg$ starke Negation
$\Rightarrow$	$\rightarrow$ starke Implikation
$\Leftrightarrow$	$\leftrightarrow$ starke Äquivalenz

Abbildung 2.18: Korrespondenz zw. VDM und $\mathcal{L}_3$

2.4.3 Mehrwertige Schaltkreise

Mehrwertige Logiken wurden auf dem Gebiet elektronischer Schaltkreise schon recht früh und für die unterschiedlichsten Zwecke eingesetzt. Eine der frühesten Arbeiten, die einen dritten Wahrheitswert in der Modellierung von Schaltkreisen benutzt, ist [Muller, 1959].

Dreiwertige Simulation von Schaltkreisen [Breuer, 1972][Hayes, 1986a] [Jephson *et al.*, 1969][Pomeranz & Reddy, 1995].

Weitere Literatur: [Epstein, 1993], [Muzio & Wesselkamper, 1986], [Smith & Glulak, 1993], [Hayes, 1986b].

Hazard detection [Yoeli & Rinon, 1964, Eichelberger, 1965]

Testmustergenerierung

Wir erklären zunächst kurz die Problemstellung. Eine ausführliche Darstellungen finden man in dem Buch [Fujiwara, 1986]. Wir reden hier von Test, die an gefertigten Chips durchgeführt werden. Gegeben ist ein Chip, der einen auf Gatterebene vorgegeben Schaltkreis realisiert. Die Korrektheit des Schaltkreises steht in dieser Phase nicht mehr zur Diskussion. Hier sollen Fehler, die im physikalischen Fertigungsprozeß aufgetreten sein können, entdeckt werden. Man geht dabei von der Annahme aus, daß der Chip nur einen einzigen Fehler enthält (single fault hypothesis). Diese Annahme ist zunächst gerechtfertigt durch die empirischen Daten, daß relativ selten mehr als ein Fertigungsfehler auf einem Chip auftritt. Zum anderen macht man mit dieser Annahme auch die Not zu einer Tugend, dann die Analyse von zwei unabhängigen Fehlern auf einem Chip würde unvergleichlich mehr Aufwand erfordern.

Die Phase der Fehlermodellierung geht der Testmustergenerierung voran. Dabei identifiziert man Fehlertypen, die realistischerweise auftreten können und deren Vorhandensein oder Abwesenheit man testen möchte. Zur Illustration betrachten wir den Schaltkreis C_1 aus Abbildung 2.19, der auch in [Fujiwara, 1986] als Beispiel benutzt wird. Ein Fehler der häufig beobachtet werden

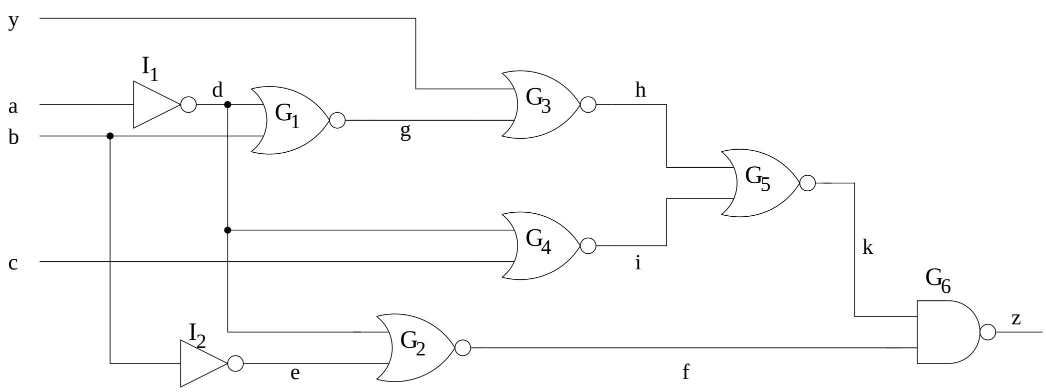


Abbildung 2.19: Beispiel eines kombinatorischen Schaltkreises C_1

kann, besteht darin, daß die Realisierung eines Gatters stets denselben Ausgangswert 0 oder 1 liefert, unabhängig von wechselnden Eingaben. Im Englischen sagt man *stuck at zero* oder *stuck at one*. In unserem Beispiel wollen

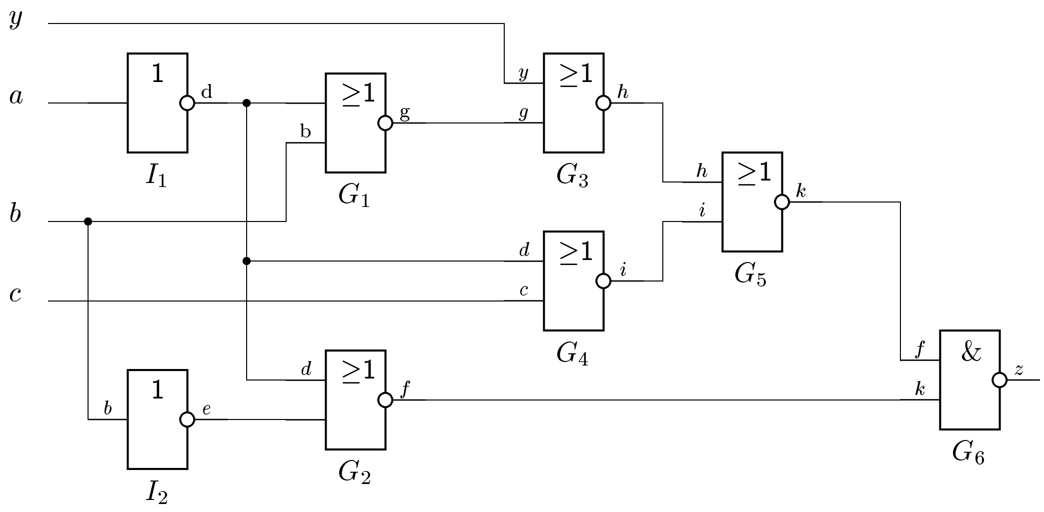


Abbildung 2.20: Schaltkreises C_1 in neuen Symbolen

wir den Fehler entdecken, daß der Inverter I_1 konstant den Wert 1 ausgibt. Wir suchen also einen Belegung der Eingänge a, b, c, y mit Booleschen Werten $(v_a, v_b, v_c, v_y) \in \{0, 1\}^4$ und einen Wert v, so daß bei korrektem Funktionieren des Chips und der Belegung der Eingänge mit (v_a, v_b, v_c, v_y) am Ausgang z der Wert v beobachtet wird. Tritt jedoch die genannte Fehlfunktion von I_1 auf, so produziert die Eingabe (v_a, v_b, v_c, v_y) am Ausgang z das Ergebnis $\bar{v}$, i.e. den Komplementärwert von v. Das Tupel (v_a, v_b, v_c, v_y) heißt ein *Testmuster* für den Fehler, daß I_1 konstant den Wert 1 liefert. Wir skizzieren, anhand unseres Beispiels, einen Verfahren zum Finden von Testmustern. In [Fujiwara, 1986] wird dieses Verfahren D-Algorithmus genannt.

In Phase 1 ist eine Belegung der Eingangsvariablen zu finden, so daß die Wert der Leitung d komplementär zur Fehlersituation wird. Das ist in Schaltkreis C_1 einfach: a muß auf 1 gesetzt werden. Leitung d hat dann in der Fehlersituation den Wert 1, bei korrektem Funktionieren den Wert 0.

Um diese Information in kompakter Form notieren und verarbeiten zu können, benutzen wir eine vierwertige Logik mit den Wahrheitswerten $\{1, 0, D, \bar{D}\}$. Einer Variablen $v \in \{a - k, y, z\}$ ordnen wir den Wert 1(0) zu, wenn v sowohl im korrekten, als auch im fehlerhaften Schaltkreis den Wert 1 (0) hat. Der Wert D wird zugewiesen, wenn im fehlerfreien Fall v den Wert 1 und andernfalls den Wert 0 hat. Entsprechend hat v den Wert $\bar{D}$, wenn im

NAND	0	1	D	$\bar{D}$
0	1	1	1	1
1	1	0	$\bar{D}$	D
D	1	$\bar{D}$	$\bar{D}$	1
$\bar{D}$	1	D	1	D

NOR	0	1	D	$\bar{D}$
0	1	0	$\bar{D}$	D
1	0	0	0	0
D	$\bar{D}$	0	$\bar{D}$	0
$\bar{D}$	D	0	0	D

X	0	1	D	$\bar{D}$
NOT X	1	0	$\bar{D}$	D

Abbildung 2.21: Wahrheitstafeln vierwertigen Logik $\mathcal{L}_D^4$

fehlerfreien Fall der Wert 0 auftritt und sonst 1. Diese Logik, wir wollen sie $\mathcal{L}_D^4$ nennen, wurde eingeführt in [Muth, 1976] (siehe auch [Fujiwara, 1986]). Aus der angegebenen Bedeutung der vier neuen Wahrheitswerte lassen sich die Wahrheitstabellen für alle Gatteroperationen ausrechnen. Für die in unserem Beispiel vorkommenden Operationen sind die Tabellen in Abb. 2.21 zusammengefaßt.

Nach der ersten Phase liegt somit folgende Variablenbelegung vor

a	b	c	d	e	f	g	h	i	k	y	z
1	–	–	$\bar{D}$	–	–	–	–	–	–	–	–

In Phase 2 des D-Algorithmus wird von d ausgehend ein Pfad zu dem einzigen Ausgang, z, gesucht, so daß auch dort ein Unterschied zwischen fehlerfreiem und fehlerbehaftetem Schaltkreis beobachtbar ist. Ein Pfad von d zum Ausgang z kann alternativ über eines der drei Gatter G_1, G_2, G_4 erfolgen. Versuchen wir zuerst den Weg über G_1. Damit am Ausgang, g, von G_1 D auftritt, muß man $b = 0$ setzen. Daraus ergib sich weiter $e = 1$, $f = 0$ und $z = 1$. Das hilft uns nicht. Nehmen wir einen zweiten Anlauf und versuchen einen Pfad von d nach z via G_2. Damit am Ausgang von G_2 D auftritt, muß $e = 0$ gelten. Daraus ergeben sich zwangsläufig $b = 1$, $g = 0$. Ingesamt also

a	b	c	d	e	f	g	h	i	k	y	z
1	1	–	$\bar{D}$	0	D	0	–	–	–	–	–

Im auch am Ausgang z einen Unterschied zwischen korrektem und fehlerhaftem Funktionieren beobachten zu können, gibt es zwei Möglichkeiten für die Belegung von k, $k = 1$ oder $k = D$. Probieren wir $k = 1$. Daraus ergeben sich zwangsläufig $h = i = 0$, $y = 1$

Ingesamt also

a	b	c	d	e	f	g	h	i	k	y	z
0	1	1	$\bar{D}$	0	D	0	0	0	1	1	$\bar{D}$

Wir haben somit ein Testmuster für den *stuck-at-one*-Fehler von I_1 gefunden.

Der D-Algorithmus ist in Wirklichkeit etwas komplizierter als die obige Kurzdarstellung erahnen läßt. Das Phänomen des *don't-care-indeterminismus* tritt wegen der Einfachheit des Beispiels nicht auf. Es kommt uns hier nicht darauf an, den D-Algorithmus im Detail vorzustellen. Wichtiger ist die Beobachtung, was dieser Algorithmus eigentlich berechnet.

Phase 1 ist in unserem Beispiel ebenfalls irreführend einfach, da das als fehlerhaft angenommene Gatter direkt nach der Eingabe vorkommt. Hätte man einen Fehler modelliert, der nahe an einem Ausgang auftritt, dann führt Phase 1 zu einem Erfüllbarkeitsproblem in der zweiwertigen Aussagenlogik.

Beobachtung: Zu jedem Schaltkreis C und jedem Fehlermodell FM gibt es eine Menge aussagenlogischer Formeln $\Delta = \Delta_{C,FM}$ in der vierwertigen Logik $\mathcal{L}_D^4$, so daß der D-Algorithmus eine erfüllende Belegung für Δ berechnet.

$$
\begin{array}{rcl@{\qquad\qquad}rcl}
d & = & \neg a & h & = & \neg(y \vee g) \\
e & = & \neg b & i & = & \neg(d \vee c) \\
f & = & \neg(d \vee e) & k & = & \neg(h \vee i) \\
g & = & \neg(d \vee b) & z & = & \neg(f \wedge k)
\end{array}
$$

Abbildung 2.22: Beschreibung des Schaltkreises C_1 aus Abb. 2.19 durch Gleichungen

Wir skizzieren anhand des oben betrachteten Beispiels, wie man die Menge Δ erhält. Zunächst besorgt man sich eine Beschreibung des Schaltkreises durch Boolesche Gleichungen, siehe Abb.2.22. Zur Notation der $\mathcal{L}_D^4$-Formeln in Δ benutzen wir Vorzeichenformeln.

Δ besteht aus

- den Gleichungen des Schaltkreises, die jetzt in der vierwertigen Logik interpretiert werden, und in Vorzeichennotation aufgeschrieben werden.

$$
\begin{array}{llll}
\{0\} & a & & \\
\{1\} & e & = & \neg b \\
\{1\} & f & = & \neg(d \vee e) \\
\{1\} & g & = & \neg(d \vee b) \\
\{1\} & h & = & \neg(y \vee g) \\
\{1\} & i & = & \neg(d \vee c) \\
\{1\} & k & = & \neg(h \vee i) \\
\{1\} & z & = & \neg(f \wedge k)
\end{array}
\qquad\qquad
\begin{array}{l}
\{1,0\}b \\
\{1,0\}c \\
\{1,0\}y \\
\{D,\bar{D}\}z \\
\{\bar{D}\}d
\end{array}
$$

Abbildung 2.23: $\mathcal{L}_D^4$-Gleichungen zur Testgenerierung

- $\{1,0\}a$ für jede Eingabevariable a
- $\bigvee_{z \in AV}\{D,\bar{D}\}z$ wobei AV die Menge aller Ausgabevariablen ist.
- die Festlegung der Eingabevariablen, wie sie als Ergebnis der ersten Phase ermittelt wurden.

Die Formel Δ für das oben betrachtet Beispiel ist in Abb. 2.23 zu sehen.

Bisher haben wir nur über kombinatorische Schaltkreise geredet. Betrachten wir jetzt das Beispiel eines sequentiellen Schaltkreises in Abb.2.24, der durch eine einfache Modifikation aus dem Schaltkreis in 2.19 entsteht. Signal y ist jetzt keine Eingabe mehr sondern das über das Verzögerungsglied D rückgekoppelte Signal k. Man kann versuchen, denselben Algorithmus anzuwenden auf die Ausfaltung des sequentiellen Schaltkreises in einen kombinatorischen. Man wird dabei allerdings feststellen, daß man nicht immer zu Ziel kommt. Auch in dem Beispiel aus Abb.2.24 scheitert der D-Algorithmus. Man kann jedoch zeigen, daß durch den Übergang zu einer neunwertigen Logik $\mathcal{L}_D^9$ wieder garantiert werden kann, daß stets ein Testmuster gefunden wird. Die Wahrheitstafel für $\mathcal{L}_D^9$ sind Abb. 2.25 zu finden.

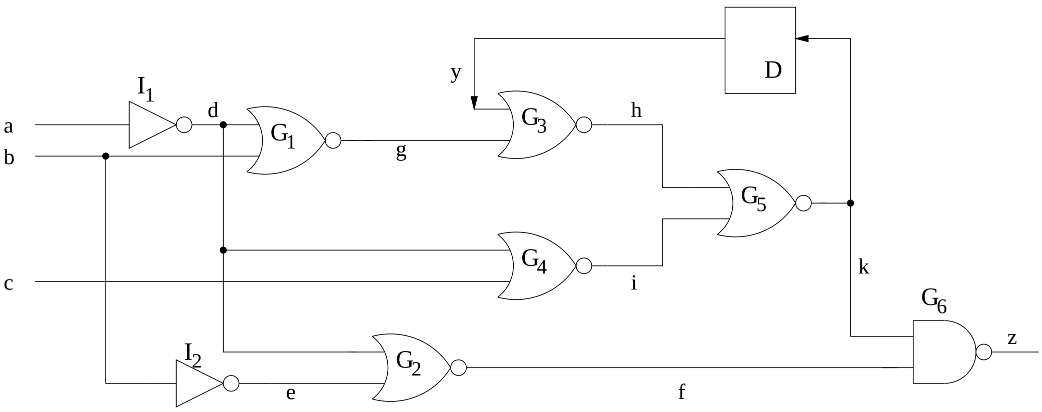


Abbildung 2.24: Beispiel eines kombinatorischen Schaltkreises C_1

und	0	$0/x$	$\bar{D}$	$x/0$	x	$x/1$	D	$1/x$	1
0	0	0	0	0	0	0	0	0	0
$0/x$	0	$0/x$	$0/x$	0	$0/x$	$0/x$	0	$0/x$	$0/x$
$\bar{D}$	0	$0/x$	$\bar{D}$	0	$0/x$	$\bar{D}$	0	$0/x$	$\bar{D}$
$x/0$	0	0	0	$x/0$	$x/0$	$x/0$	$x/0$	$x/0$	$x/0$
x	0	$0/x$	0	$x/0$	x	x	$x/0$	x	x
$x/1$	0	$0/x$	$\bar{D}$	$x/0$	x	$x/1$	$x/0$	x	$x/1$
D	0	0	0	$x/0$	$x/0$	$x/0$	D	D	D
$1/x$	0	$0/x$	$0/x$	$x/0$	x	x	D	$1/x$	$1/x$
1	0	$0/x$	$\bar{D}$	$x/0$	x	$x/1$	D	$1/x$	1

oder	0	$0/x$	$\bar{D}$	$x/0$	x	$x/1$	D	$1/x$	1
0	0	$0/x$	$\bar{D}$	$x/0$	x	$x/1$	D	$1/x$	1
$0/x$	$0/x$	$0/x$	$\bar{D}$	x	x	$0/x$	$1/x$	$1/x$	1
$\bar{D}$	$\bar{D}$	$\bar{D}$	$\bar{D}$	$x/1$	$x/1$	$x/1$	1	1	1
$x/0$	$x/0$	x	$x/1$	$x/0$	x	$x/1$	D	$1/x$	1
x	x	x	$x/1$	x	x	$x/1$	$1/x$	$1/x$	1
$x/1$	$x/1$	$x/1$	$x/1$	$x/1$	$x/1$	$x/1$	1	1	1
D	D	$1/x$	1	D	$1/x$	1	D	$1/x$	1
$1/x$	$1/x$	$1/x$	1	$1/x$	$1/x$	1	$1/x$	$1/x$	1
1	1	1	1	1	1	1	1	1	1

W	0	$0/x$	$\bar{D}$	$x/0$	x	$x/1$	D	$1/x$	1
not W	1	$1/x$	D	$x/1$	x	$x/0$	$\bar{D}$	$0/x$	0

Abbildung 2.25: Wahrheitstafel der neunwertigen Logik $\mathcal{L}_D^9$ von Muth

2.4.4 Shape Analysis

Shape Analysis is an important and well covered part of static program analysis, see e.g. [Nielson *et al.*, 1999][Section 2.9]. The central role in shape analysis is played by the set U of abstract stores, i.e. U is perceived as the abstraction of the locations program variables can *point to*. In an object-oriented context you could view U as an abstraction of the set of all objects existing at a *snapshot* during program execution. In the investigations to follow we will use U as the universe of the structure $\mathcal{S}$ that describes the abstract state of a program at a given snapshot. When necessary, we will write $U_{\mathcal{S}}$ instead of just U. For every program variable x there will be a unary predicate, also denoted by x, defined on $\mathcal{S}$. For a store $v \in U$ the intention is that $x(v)$ is true (in symbols $\mathcal{S} \models x[v]$) if x points to v. For any abstract state $\mathcal{S}$ and any program variable x we require that the unary predicate x holds true of at most one store, i.e. we require $\mathcal{S} \models \forall s_1 \forall s_1((x(s_1) \wedge x(s_2)) \rightarrow s_1 = s_2)$. It is thus possible that x does not point to any store, i.e. $\mathcal{S} \models \forall s(\neg x(s))$. Alternatively, one could have stipulated that for every abstract state $\mathcal{S}$ there is an element $null \in U$ and $\mathcal{S} \models x(null)$ signifies that x points to the *null* object. There are no essential advantages of one way over the other. The approach we have chosen leads to slightly simpler diagrams. Additional predicates on $\mathcal{S}$ depend on the considered program and the task at hand. We will, e.g. consider below a binary predicate *next*.

Properties one might be interested in could be

$\exists s x(s)$	x does not point to *null*
$\forall s(\neg(x(s) \wedge t(s)))$	x and t do not point to the same store
$\exists s\ is(s)$	*next* is a shared node

In this listing we have used the abbreviation, which will occur over and again during the rest of this subsection:

$$is(s) \equiv \exists s_1 \exists s_2(next(s_1, s) \wedge next(s_2, s) \wedge s_1 \neq s_2)$$

The goal is to prove for a given program, or a given program part, that a certain property holds at every program state, or every stable program state. A typical example that you will find in almost any introductory text on shape analysis is the preservation of cycle-freeness of a list pointer structure by the algorithm reversing the list. If n is the store representing the head of the list and we know

1. $\neg\exists v(next(v, n)$
2. $\forall v \forall w(next(m, v) \wedge next(m, w) \rightarrow v = w)$ for all stores m reachable from n,
3. $\neg\, is(m)$ for all stores m reachable from n

then the list with entry point n cannot be cyclic. We concentrate here on showing the preservation of the formula $is(s)$. Let us look at the program in

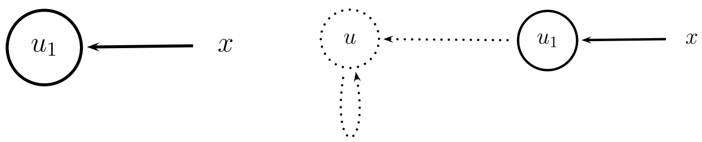
```
class ReverseList {
    int value;
    ReverseList next;

 public ReverseList reverse() {
        ReverseList t, y= null, x = this;
        while (x != null) {
        st1:    t=y;
        st2:    y=x;
        st3:    x=x.next;
        st4:    y.next = t;}
        return y;}}
```

Abbildung 2.26: A program reversing a list

Figure 2.26 (an executable version of this program can be found in Section 6.1). The labels `sti` have been added for future reference. We set ourselves the following task

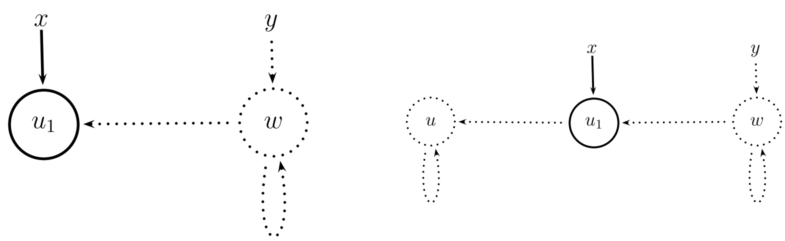
> Assume that at the beginning of the `while` loop $\mathcal{S} \models \neg is(n)$ is true for all stores n in the list. Show that in the state $\mathcal{S}_e$ after execution of the `while` loop again $\mathcal{S}_e \models \neg is(n)$ holds true for all n.

Since we cannot make any assumptions on the set of stores U at the start of the `while`-loop we need to investigate infinitely many structures, which obviously is not possible. We will present here an attempt to tackle this problem using three-valued logic that has been proposed by Mooly Sagiv, Thomas Reps and Reinhard Wilhelm. These three authors have published quite a few papers on this issue. We will closely follow their technical report [Sagiv *et al.*, 2000]. We will demonstrate the approach by way of example using the program *reverse* from Figure 2.26 and the property $\forall s \neg is(s)$. From this the reader should be able to pick up the general idea. The main incredient in the solution is the use of three-valued structures to approximate two-valued structures. More precisely, we try to find finitely many three-valued structures $\mathcal{S}^3_1, \ldots, \mathcal{S}^3_k$ such that for an arbitrary two-valued abstract state $\mathcal{S}$ that may be possible before the `while`-loop starts there is a surjective mapping F from $\mathcal{S}$ onto one of the $\mathcal{S}^3_i$ for $1 \leq i \leq k$ with $\mathcal{S} \sqsubseteq^F \mathcal{S}^3_i$. (see Definition 16 on page 36 for an explanation of the relation $\sqsubseteq^F$) In the terminology of that definition we thus have that every possible initial state has an abstraction among $\mathcal{S}^3_1, \ldots, \mathcal{S}^3_k$.



(a) $\mathcal{S}^3_1$ (b) $\mathcal{S}^3_2$

Abbildung 2.27: Considered three-valued base structures



(a) $\mathcal{S}^3_{1+}$ (b) $\mathcal{S}^3_{2+}$

Abbildung 2.28: Excluded three-valued base structures

We will follow the following master plan:

1. For every three-valued structure $\mathcal{S}_i^3$ we will define an algorithm to compute a three-valued structure $\mathcal{S}_{i,e}^3$. We think of $\mathcal{S}_{i,e}^3$ as the three-valued state reached after execution of α_r , the body of the `while`-loop, when started in $\mathcal{S}_i^3$. This needs further explanation. If $\mathcal{S}$ is a two-valued state it is fairly straight forward to compute the two-valued state $\mathcal{S}_e$ that is reached after executing α_r starting with $\mathcal{S}$, since the commands in α_r are so simple. The construction of $\mathcal{S}_{i,e}^3$ will be done such that $\mathcal{S} \sqsubseteq^F \mathcal{S}_i^3$ implies $\mathcal{S}_e \sqsubseteq^F \mathcal{S}_{i,e}^3$. This may sound difficult, but it will turn out to be rather easy.

2. Determine a set $\mathcal{M}_0$ of abstract three-valued states to start with.

3. At iteration k ($k \geq 1$) we are dealing with a set $\mathcal{M}_{k-1}$ of abstract three-valued states. We try to prove for every $\mathcal{S}^3 \in \mathcal{M}_{k-1}$ that $v_{\mathcal{S}^3}(\forall s(\neg is(s))) = 1$ implies $v_{\mathcal{S}_e^3}(\forall s(\neg is(s))) = 1$, then Exercise 2.1.10 will help us to conclude for any two-valued state $\mathcal{S}$ that is reachable with $k-1$ iterations of α_r:

   $$\mathcal{S} \models \forall s \neg is(s) \Rightarrow \mathcal{S}_e \models \forall s \neg is(s)$$

   If we succeed we set

   $$\mathcal{M}_k = \{\mathcal{S}_e^3 \mid \mathcal{S}^3 \in \mathcal{M}_{k-1}\}$$

   If $\mathcal{M}_k \subseteq \mathcal{M}_{k-1}$ we are finished and the claim is positively established. Otherwise we repeat step 3 with $\mathcal{M}_k$.

   If we get for one $\mathcal{S}^3 \in \mathcal{M}_{k-1}$ $v_{\mathcal{S}_e^3}(\forall s(\neg is(s))) = 0$ then our conjecture was false.

   If we get for one $\mathcal{S}^3 \in \mathcal{M}_{k-1}$ $v_{\mathcal{S}_e^3}(\forall s(\neg is(s))) = \frac{1}{2}$ then we are lost, this result is inconclusive. Should this happen we would iterate the procedure with a larger set $\mathcal{M}'_{k-1}$.

There is, unfortunately, no guarantee that this iteration will come to a conclusive end in the general case.

Figure 2.27 lists, $\mathcal{M}_0$, i.e. the three-valued structures that we want to take as the basis for our investigation. Let us first explain how to read the shown diagrams, which are meant to represent three-valued structures. Let us look at $\mathcal{S}_2^3$. Every circle denotes one element of this structure, i.e. $\mathcal{S}_2^3$ contains exactly two elements. Furthermore, for all two-valued structures $\mathcal{S}$ and mappings F with $\mathcal{S} \sqsubseteq^F \mathcal{S}^3_{2+}$ we require that for any node n in $U_{\mathcal{S}^3_{2+}}$ with a solid circle $\{v \in U_{\mathcal{S}} \mid F(v) = n\}$ has exactly one element, while for nodes n with a dotted circle $\{v \in U_{\mathcal{S}} \mid F(v) = n\}$ must have at least one element, but may otherwise be of arbitrary cardinality. This is why nodes with a dotted circle are called *summary nodes* in the parlance of shape theory. An arrow from the variable x to a node n in $\mathcal{S}^3_2$ is to be interpreted

as $v_{\mathcal{S}^3_2}(x[n]) = 1$ if it is a solid arrow,
as $v_{\mathcal{S}^3_2}(x[n]) = \frac{1}{2}$ if it is a dotted arrow and
as $v_{\mathcal{S}^3_2}(x[n]) = 0$ if there is no arrow from x to n.

For two nodes n_1, n_2 we agree on the following coding

$v_{\mathcal{S}^3_2}(next[n_1, n_2]) = 1$ if the is a solid arrow from n_1 to n_2
$v_{\mathcal{S}^3_2}(next[n_1, n_2]) = \frac{1}{2}$ if the is a dotted arrow from n_1 to n_2
$v_{\mathcal{S}^3_2}(next[n_1, n_2]) = 0$ if the is no arrow from n_1 to n_2

The same explanations apply, of course, to all the other structures $\mathcal{S}^3_{i(+)}$ in Figures 2.28 and 2.27. The three-valued structure $\mathcal{S}^3_2$, e.g. serves as an abstraction for all two-valued states in which x points to a store u_1, y and t point to *null* and there is no incoming *next*-edge into n and the list starting at u_1 consists of 2 or more elements. Structure $\mathcal{S}^3_1$ represents a one-element list starting at u_1. We need not consider the empty list here, since the guard of the loop requires that x does not point to *null*. Thus any initial structure should have ar least one element, the store x points to. The structures in Figure 2.28 are not considered in the beginning, since we assume that there a no incoming `next`-edges into u_1.

Let us now turn to the computation of $\mathcal{S}^3_e$ from $\mathcal{S}^3$. We do this by describing the effect of each statement `sti` on the state it is executed in. There are many ways to do this. We will stick to the way it is done in [Sagiv *et al.*, 2000] using *update formulas*. For the statement `st1: t=y;` the update formula is $t'(v) = y(v)$. The primed symbols refer to the predicate in the resulting structure. The unprimed symbols to the predicates in the initial structure. This is already an abbreviation in full the update formula is

$$\begin{array}{ll} \forall s_1 \forall s_2(& t'(s) = y(s) \wedge \\ & x'(s) = x(s) \wedge y'(s) = y(s) \\ & next'(s_1, s_2) = next(s_1, s_2) \quad) \end{array}$$

We will omit the parts that do not change. Here is a list of update formulas

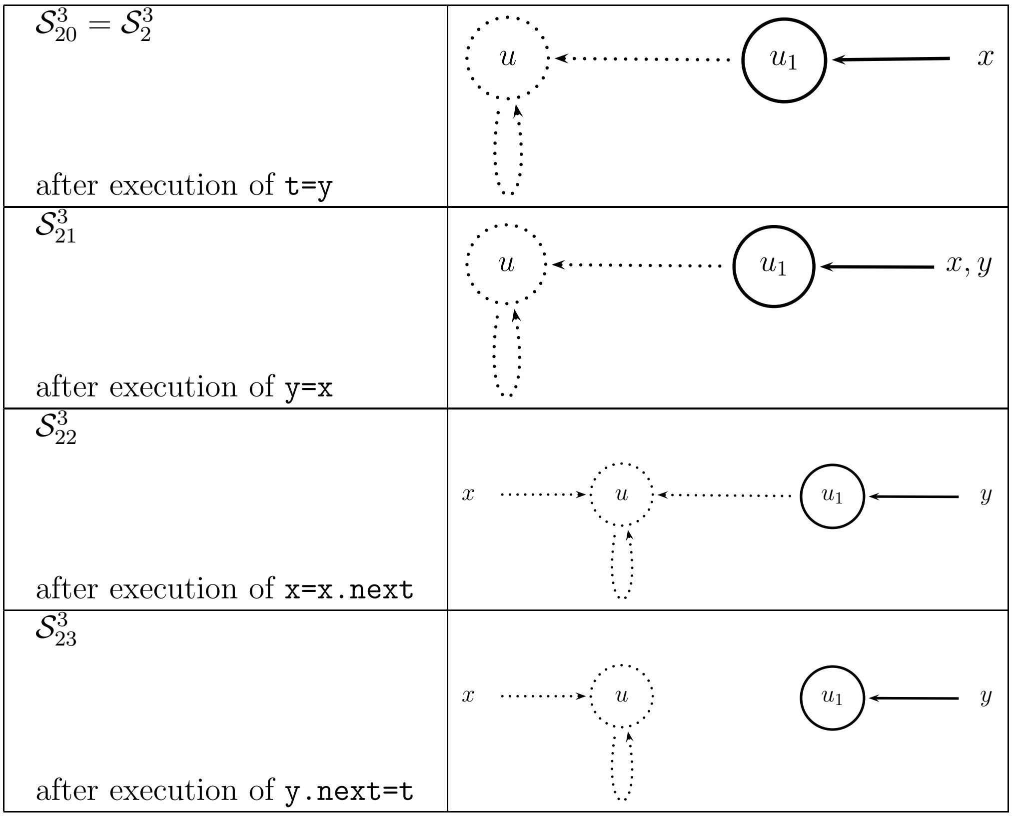


Abbildung 2.29: Executing α_r on $\mathcal{S}^3_2$

for all statements in the body α_r of the `while`-loop.

`st1: t=y;`	$t'(s) = y(s)$
`st2: y=x;`	$y'(s) = x(s)$
`st3: x=x.next;`	$x'(s) = \exists s_1(x(s_1) \wedge next(s_1, s))$
`st4: y.next=t;`	$next'(s_1, s_2) = (next(s_1, s_2) \wedge \neg y(s_1)) \vee (y(s_1) \wedge t(s_2))$

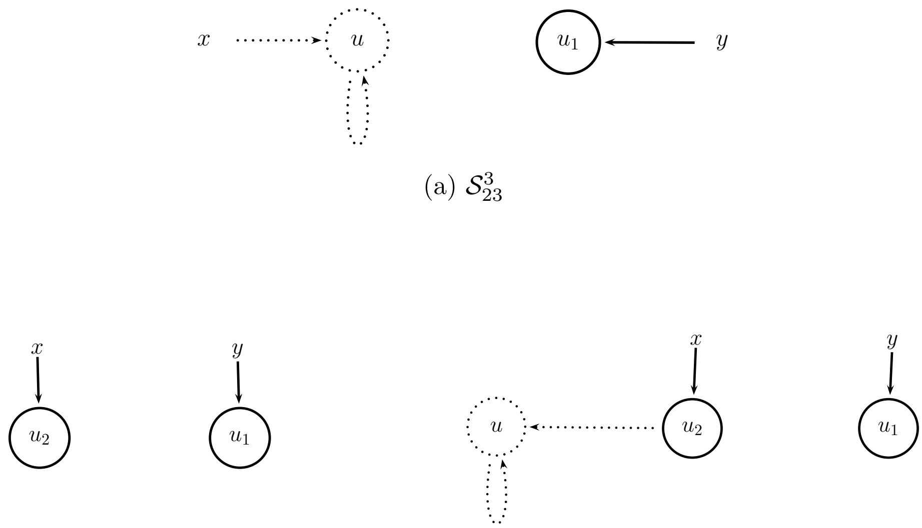
It is, of course, crucial to get the update formulas right. They incorporate the semantics of the progaming language one is dealing with. In the case at hand, finding the proper update formulas was, however, a piece of cake.

Figure 2.29 shows the three-valued states arising from $\mathcal{S}^3_2$ through the execution of α_r.

The next step is to determine how the formula $is(v)$ is effected by state changes. We again follow [Sagiv *et al.*, 2000] and also provide an update formula for the $is(s)$. From the definition of $is(s)$ is is apparent that $is'(s) = is(s)$ for the statements `st1` to `st3`. Here is the update formula for the remaining case of `st4`.

$$
\begin{aligned}
is'(s) = \;& \exists s_1, s_2(next(s_1, s) \wedge next(s_2, s) \wedge s_1 \neq s_2 \wedge \neg y(s_1) \wedge \neg y(s_2)) \\
& \vee \\
& t(s) \wedge \exists s_1(next(s_1, s) \wedge \neg y(s_1))
\end{aligned}
$$

The formula given in [Sagiv *et al.*, 2000] is different, but equivalent. They have tuned their formula for efficient computation. The formula we want to prove is thus treated in the same way as the predicates that define the state structures. This may be the reason why Sagiv and his co-authors call formulas like $is(s)$ also *instrumentation predicates.*



(a) $\mathcal{S}^3_{23}$

(b) $\mathcal{S}^3_3$

(c) $\mathcal{S}^3_4$

Abbildung 2.30: Focusing on x

Let us look at Figure 2.29 first and assume that in the initial structure we have $v_{\mathcal{S}^3_2}(\exists s\ is(s)) = 0$. It is almost obvious that we still have $v_{\mathcal{S}^3_{2e}}(\exists s\ is(s)) = 0$ in the final structure $\mathcal{S}^3_{2e}$. You could carefully evaluate the update formula for is or observe that after all, we only omited one `next`-edge.

So far we have only considered one iteration of the loop body α_r. If the final structure, $\mathcal{S}^3_{23}$ already occured among the base structures we would, of course, be finished. But, this is far from true. So we have to execute symbolically the loop body once again. Before we do this we encounter another feature of the considered method: It does not allow structures $\mathcal{S}^3$ with $v_{\mathcal{S}^3}(x(n)) = \frac{1}{2}$, for x or any other pointer predicate. We need therefore refine structure $\mathcal{S}^3_{23}$ in all possible ways that renders the value of $x(n)$ definite for all stores n. This process is called *focussing on x*.

Figure 2.30 shows the two structures $\mathcal{S}^3_3$ and $\mathcal{S}^3_4$ resulting from $\mathcal{S}^3_{23}$ by focusing on x.

Figure 2.31 shows the effects of executing α_r on $\mathcal{S}^3_4$. It is again easily checked that $v(\exists s\ is(s)) = 0$ for all structures $\mathcal{S}^3_{4i}$, $0 \leq i \leq 4$.

Figure 2.32 now shows what happens if we iterate execution of the loop body α_r four times. It is easy to see that focusing in x in the structure $\mathcal{S}^3 6e$ (last table entry in Figure 2.32) we get again $\mathcal{S}^3 6$. The nodes to the left of the node t is pointing to get absorbed into one summary node. There are of course some loose ends, that we did not follow up here, e.g. the structures without summary nodes that occur by focusing. But, these are easier case.

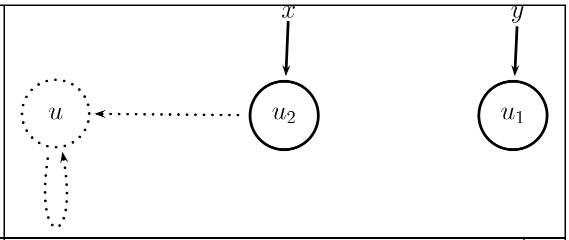
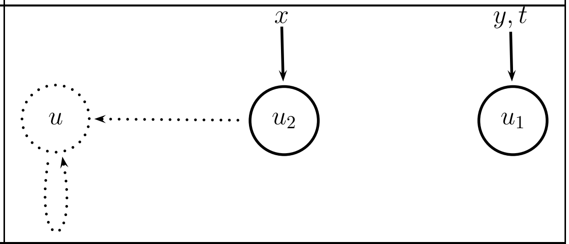
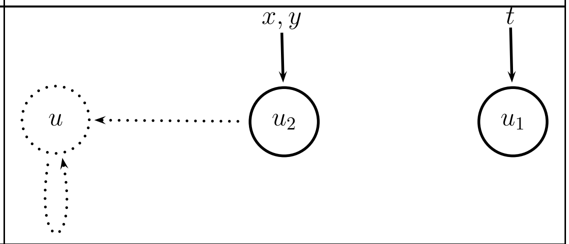
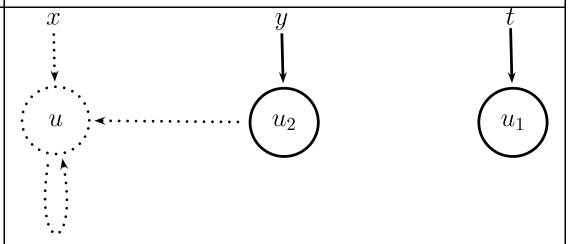
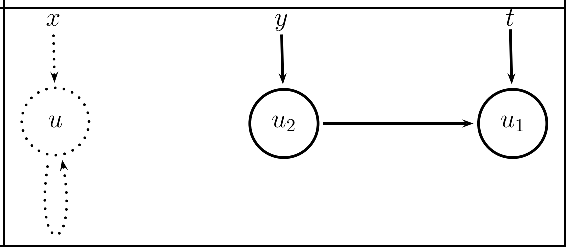
$\mathcal{S}^3_{40} = \mathcal{S}^3_4$ entering `while`-loop	
$\mathcal{S}^3_{41}$ after execution of `t=y`	
$\mathcal{S}^3_{42}$ after execution of `y=x`	
$\mathcal{S}^3_{43}$ after execution of `x=x.next`	
$\mathcal{S}^3_{44}$ after execution of `y.next=t`	

Abbildung 2.31: Executing α_r on $\mathcal{S}^3_4$

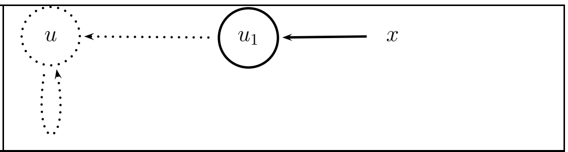
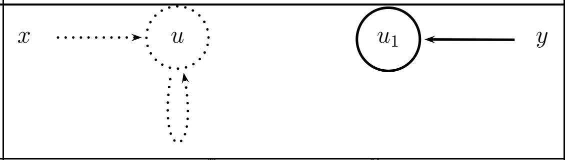
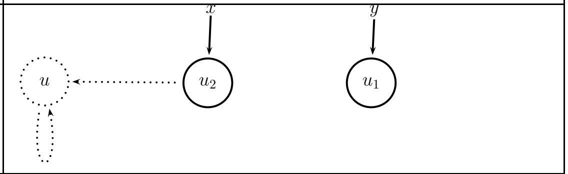
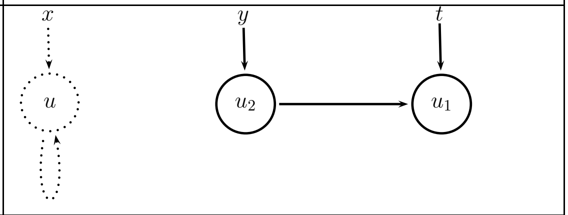
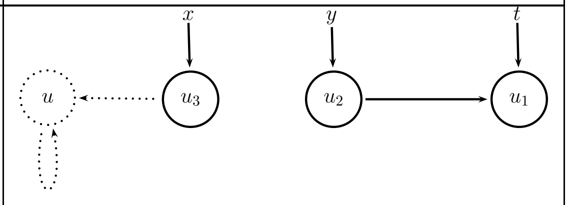
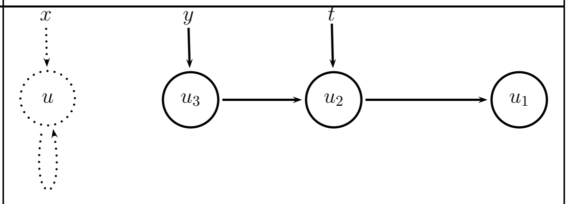
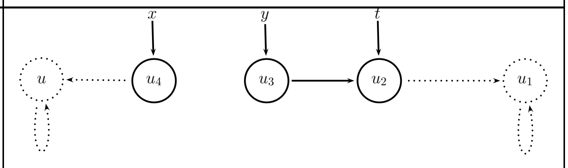
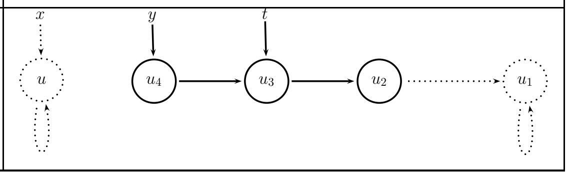
$\mathcal{S}_2^3$ before first iteration	
$\mathcal{S}_{2e}^3$ after first iteration	
$\mathcal{S}_4^3$ focusing on x	
$\mathcal{S}_{4e}^3$ after second interation	
$\mathcal{S}_5^3$ focusig on x	
$\mathcal{S}_{5e}^3$ after third iteration	
$\mathcal{S}_6^3$ focusing on x	
$\mathcal{S}_{6e}^3$ after fourth iteration	

Abbildung 2.32: Iterations of α_r on $\mathcal{S}_2^3$

2.5 Unendlichwertige Logiken

2.5.1 Łukasiewicz Logiken

Als Einstieg in unendlichwertige Logiken betrachten wir die Familie der Łukasiewizc Logiken $\mathcal{L}^n_{Łukas}$ für $n \geq 2$ und $n = \aleph_0, \aleph_1$. Die Symbole $\aleph_0, \aleph_1$ werden gelesen als aleph null, bzw. aleph eins. In der Theorie der Kardinalzahlen dienen sie üblicherweise zur Bezeichnung der Kardinalität der Menge der rationalen, bzw. der reellen Zahlen. Wir benutzen diese Zeichen hier aus rein historischen Gründen, wir hätten an ihrer Stelle genausogut zwei andere von allen natürlichen Zahlen verschiedene Symbole benutzen können. Die Menge W_n der Wahrheitswerte für $\mathcal{L}^n_{Łukas}$ ist dabei

$$\begin{array}{lcll} W_n & = & \{0, \frac{1}{n-1}, \frac{2}{n-1}, \ldots 1\} & \text{falls } n \in I\!N \\ W_n & = & \text{das rationale Intervall } [0,1] & \text{falls } n = \aleph_0 \\ W_n & = & \text{das relle Intervall } [0,1] & \text{falls } n = \aleph_1 \end{array}$$

Die aussagenlogischen Junktoren für alle $\mathcal{L}^n_{Łukas}$ sind

$$\neg, \Rightarrow, \wedge, \vee, \equiv$$

Der einzige ausgezeichnete Wahrheitswert ist 1. Anstelle von Wahrheitstabellen geben wir die Definition der durch diese Junktoren induzierten Funktionen auf W_n direkt an:

$$\begin{array}{lcl} \neg A & = & 1 - A \\ A \Rightarrow B & = & min\{1, 1 - A + B\} \\ A \vee B & = & max\{A, B\} \\ A \wedge B & = & min\{A, B\} \\ A \equiv B & = & 1 - \mid A - B \mid \end{array}$$

Diese Notation ist eine abkürzende Schreibweise, die wir dem forgeschrittenen Leser glauben zumuten zu können. Formal genau ist z.B. die zweite Zeile wie folgt zu lesen :
Ist $v : Var \rightarrow W_n$ eine Belegung der aussagenlogischen Variablen mit Wahrheitswerten aus W_n, dann gilt

$$v(A \Rightarrow B) = min\{1, 1 - v(A) + v(B)\}$$

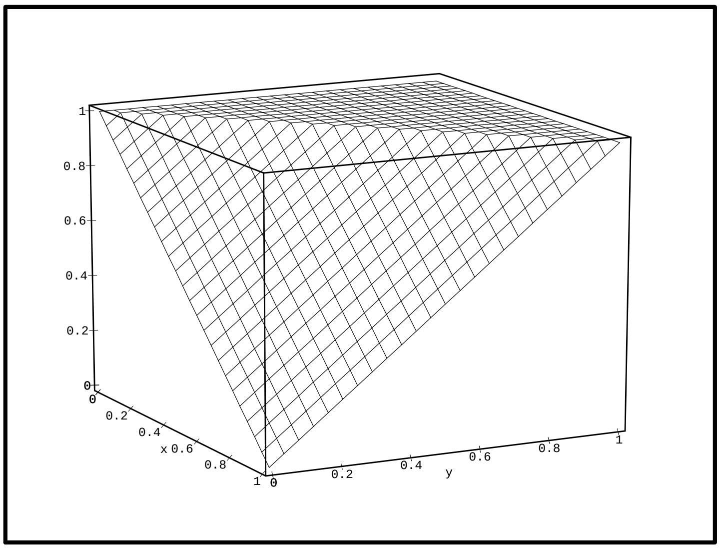


Abbildung 2.33: Łukasiewicz Implikation

Man beachte, daß diese Definition, wie die anderen auch, nicht von n abhängt, in dem Sinne, daß für alle n, m mit $W_n \subseteq W_m$ und $v : Var \rightarrow W_n$ für alle Formeln A, die Auswertung $v(A)$ in $\mathcal{L}^n_{Łukas}$ übereinstimmt mit der Auswertung $v(A)$ in $\mathcal{L}^m_{Łukas}$.
Für $n = 3$ ist dann $\neg$ die starke Negation, $\vee$ und $\wedge$ stimmen mit der Interpretation in $\mathcal{L}_3$ überein und die drei-wertige Wahrheitstabelle für $\Rightarrow$ wurde schon in den Abbildungen 2.12 und 2.2 gezeigt. Der Werteverlauf der Implikation über der Wertemenge $W_{\aleph_1}$ ist in Abbildung 2.33 dargestellt.
Mit $Taut(\mathcal{L}^n_{Łukas})$ bezeichnen wir die Menge der Tautologien in $\mathcal{L}^n_{Łukas}$. Wir beginnen mit einigen einfachen Eigenschaften der Tautologien in $\mathcal{L}^n_{Łukas}$.

Lemma 19

1. *Für $n, m \in I\!N$, sodaß $(m-1)$ ein Teiler von $(n-1)$ ist, gilt*

$$Taut(\mathcal{L}^n_{Łukas}) \subseteq Taut(\mathcal{L}^m_{Łukas})$$

   Die Umkehrung dieser Aussage wird später bewiesen.

2. $Taut(\mathcal{L}^{\aleph_0}_{Łukas}) = Taut(\mathcal{L}^{\aleph_1}_{Łukas})$

3. $Taut(\mathcal{L}^{\aleph_0}_{Łukas}) = \bigcap\{Taut(\mathcal{L}^n_{Łukas}) \mid n \geq 2, n \in I\!N\}$

Beweis:
(1) Sei $m-1$ ein Teiler von $n-1$, etwa $(n-1) = (m-1) \cdot t$ mit $t > 1$. Dann gilt sicherlich $W_m \subseteq W_n$, denn jedes Element $\frac{k}{m-1} \in W_m$ kommt auch als $\frac{k \cdot t}{(m-1) \cdot t}$ in W_n vor. Ist A eine Formel der Łukasiewicz Logik mit den Aussagenvariablen $p_1, \ldots, p_s$ so bezeichnen wir mit A_n die von A induzierte Funktion auf W_n^s und entsprechend mit A_m die von A auf W_m^s induzierte Funktion. Man sieht leicht aus der Interpretation der aussagenlogischen Operatoren, daß für alle $w_1, \ldots, w_s \in W_m$ gilt:

$$A_n(w_1, \ldots, w_s) = A_m(w_1, \ldots, w_s)$$

(2) Die Inklusion $Taut(\mathcal{L}^{\aleph_1}_{Łukas}) \subseteq Taut(\mathcal{L}^{\aleph_0}_{Łukas})$ gilt offensichtlich. Sei jetzt H eine Formel mit k Aussagenvariablen und eine Tautolgie in $\mathcal{L}^{\aleph_0}_{Łukas}$, d.h. für jedes k-Tupel $\vec{r}$ rationaler Zahlen gilt $H(\vec{r}) = 1$. Man überzeugt sich durch Induktion über den Aufbau der Formel H, daß H eine stetige Funktion von dem reellen Intervall $[0,1]^k$ in $[0,1]$ induziert. Für ein reelles Tupel $\vec{r} \in [0,1]^k$, wähle man eine Folge $\vec{r_i}$ rationaler Tupel mit $\lim_i \vec{r_i} = \vec{r}$ und es folgt $H(\vec{r}) = \lim_i H(\vec{r_i}) = \lim_i 1 = 1$.
(3) Die Inklusionen $Taut(\mathcal{L}^{\aleph_0}_{Łukas}) \subseteq Taut(\mathcal{L}^{n}_{Łukas})$ für alle $n \in \mathbb{N}$ gelten offensichtlich und damit auch $Taut(\mathcal{L}^{\aleph_0}_{Łukas}) \subseteq \bigcap\{Taut(\mathcal{L}^{n}_{Łukas}) \mid n \geq 2, n \in I\!N\}$. Sei jetzt F eine Formel mit den aussagenlogischen Variablen $p_1, \ldots, p_r$, die nicht in $Taut(\mathcal{L}^{\aleph_0}_{Łukas})$ liegt. Dann gibt es eine Belegungsfunktion v mit $v(F) \neq 1$ und $v(p_i) \in \mathbb{Q}$ für alle $1 \leq i \leq r$. Ist n ein gemeinsamer Nenner aller Brüche $v(p_i)$, dann liegen alle diese Werte schon in W_{n+1}. Also gilt auch $F \notin Taut(\mathcal{L}^{n}_{Łukas})$.

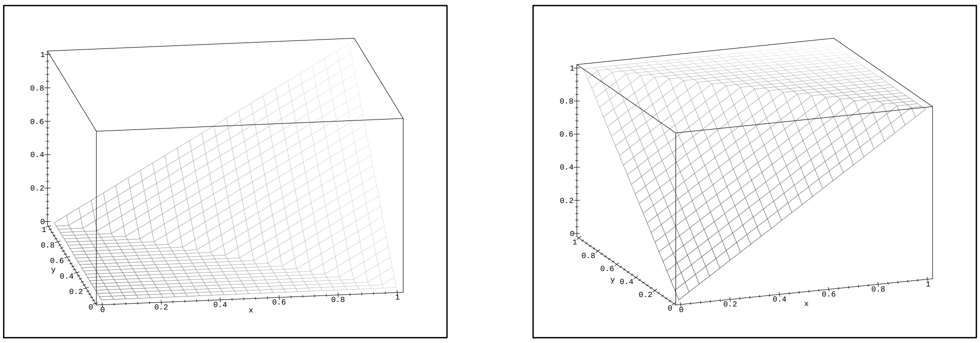
■

Von besonderem Interesse in den Łukasiewicz Logiken sind die Operatoren $\sqcup$ (*truncated sum,abgeschnittene Summe*), $\sqcap$ (*truncated product,abgeschnittenes Produkt*).

Definition 28

$$\begin{array}{lclcl} A \sqcup B & = & \neg A \Rightarrow B & = & min\{1, A+B\} \\ A \sqcap B & = & \neg(\neg A \sqcup \neg B) & = & max\{0, A+B-1\} \end{array}$$

siehe Abbildung 2.34



$A \sqcap B$ $\qquad\qquad$ $A \sqcup B$

Abbildung 2.34: Die logischen Operatoren aus Definition 28

Mit $\bigsqcup^m A$ bezeichnen wir die m-fache Iteration von $\sqcup$, also

$$\bigsqcup^m A = \underbrace{A \sqcup A \sqcup \ldots \sqcup A}_{m \text{ mal}}$$

und dual

$$\bigsqcap^m A = \underbrace{A \sqcap A \sqcap \ldots \sqcap A}_{m \text{ mal}}$$

Lemma 20 *Es gilt:*

1. $\bigsqcup^m A = min(1, mA)$
2. $\bigsqcap^m A = 1 - min(1, m(1 - A))$

Beweis:
zu 1: Induktion nach m. Der Induktionsanfang $m = 1$ ist trivialerweise wahr. Betrachten wir den Schritt von m auf $m + 1$

$$\begin{array}{lcll} \bigsqcup^{m+1} A & = & A \sqcup \bigsqcup^m A & \\ & = & min(1, A + \bigsqcup^m A) & \text{Def. von } \sqcup \\ & = & min(1, A + min(1, mA)) & \text{Ind.Hyp.} \end{array}$$

Man sieht leicht

$$min(1, A + min(1, mA)) = \begin{cases} 1 & \text{falls } 1 \leq mA \\ min(1, (m+1)A) & \text{falls } mA < 1 \end{cases}$$

Da im Fall $1 \leq mA$ auch $min(1, (m + 1)A) = 1$ gilt haben wir insgesamt

$$min(1, A + min(1, mA)) = min(1, (m + 1)A)$$

zu 2: Folgt unmittelbar aus 1. und dr Definition $A \sqcap B = \neg(\neg A \sqcup \neg B)$.

■

Lemma 21

1. $$\bigsqcup^{m} A \Rightarrow \bigsqcup^{m-1} A \in Taut(\mathcal{L}^n_{Łukas})$$
   genau, dann wenn $n \leq m$.

2. *Aus $Taut(\mathcal{L}^n_{Łukas}) \subseteq Taut(\mathcal{L}^m_{Łukas})$ folgt $n \geq m$.*

3. *In $\mathcal{L}^n_{Łukas}$ gilt für jede Belegung $v : AL_{var} \to W_n$ der aussagenlogischen Variablen*
   $$v(\bigsqcup^{n-1} \neg A) = \begin{cases} 0 & \textit{falls } v(A) = 1 \\ 1 & \textit{falls } v(A) < 1 \end{cases}$$

4. *Für die Formel*
   $$C_{m-1} = (\bigsqcup_{i=1}^{m-1} \neg A) \wedge (A \sqcup \textstyle\bigsqcap_{i=1}^{m-2} A)$$
   gilt
   $$v(C_{m-1}) = 1 \textit{ gdw } v(A) = \frac{m-2}{m-1}$$

Beweis:
zu (1):

$$\begin{array}{ll}
 & \bigsqcup^m A \Rightarrow \bigsqcup^{m-1} A = 1 \text{ in } \mathcal{L}^n_{Łukas} \\
\text{gdw} & min\{1, 1 - min\{1, m \cdot A\} + min\{1, (m-1) \cdot A\}\} = 1 \\
\text{gdw} & min\{1, m \cdot A\} \leq min\{1, (m-1) \cdot A\} \\
\text{gdw} & (m-1) \cdot A \geq 1 \text{ für alle } A \in W_n \text{ mit } A > 0 \\
\text{gdw} & A \geq \frac{1}{m-1} \text{ für alle } A \in W_n \text{ mit } A > 0 \\
\text{gdw} & n \leq m
\end{array}$$

Siehe auch Abbildung 2.35

zu (2):

Gilt $n < m$, dann gilt $(\bigsqcup^n A \Rightarrow \bigsqcup^{n-1} A) \in Taut(\mathcal{L}^n_{Łukas})$ und $(\bigsqcup^n A \Rightarrow \bigsqcup^{n-1} A) \notin Taut(\mathcal{L}^m_{Łukas})$

zu (3):

$$\begin{array}{ll}
 & v(\bigsqcup_{i=1}^{n-1} \neg A) = 1 \\
\text{gdw} & min\{1, (n-1) \cdot (1 - v(A))\} = 1 \\
\text{gdw} & (n-1) \cdot (1 - v(A)) \geq 1 \\
\text{gdw} & v(A) \leq \frac{n-2}{n-1}
\end{array}$$

Da für jedes $p \in W_n$ mit $p < 1$ schon $p \leq \frac{n-2}{n-1}$ gilt, folgt die Behauptung.

zu (4):

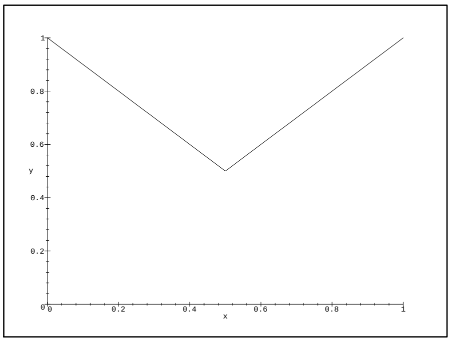
Da $v(C_{m-1}) = 1$ genau dann gilt, wenn $v(\bigsqcup_{i=1}^{m-1} \neg A) = 1$ und $v(A \sqcup \bigsqcap_{i=1}^{m-2} A) = 1$ gilt, betrachten wir die beiden Teilformeln getrennt.

Zunächst gilt nach (3) $v(\bigsqcup_{i=1}^{m-1} \neg A) = 1$ genau dann, wenn $v(A) \leq \frac{m-2}{m-1}$
Die Untersuchung der zweiten Teilformel liefert

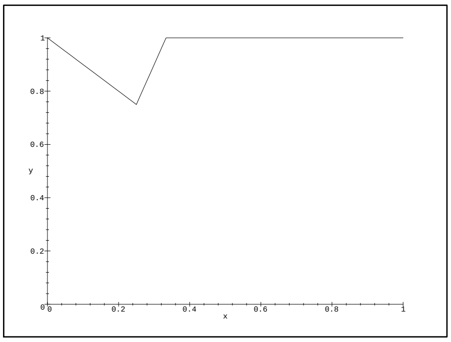
$$\begin{array}{ll}
 & v(A \sqcup \bigsqcap_{i=1}^{m-2} A) = 1 \\
\text{gdw} & min\{1, v(A) + v(\bigsqcap_{i=1}^{m-2} A)\} = 1 \\
\text{gdw} & v(A) + v(\bigsqcap_{i=1}^{m-2} A) \geq 1 \\
\text{gdw} & v(A) + max\{0, (m-2)v(A) - (m-2) + 1\} \geq 1 \\
\text{gdw} & max\{v(A), (m-1)v(A) - m + 3\} \geq 1 \\
\text{gdw} & v(A) \geq 1 \text{ oder } (m-1)v(A) - m + 3 \geq 1 \\
\text{gdw} & v(A) \geq 1 \text{ oder } v(A) \geq \frac{m-2}{m-1} \\
\text{gdw} & v(A) \geq \frac{m-2}{m-1}
\end{array}$$

Zusammengenommen ergibt sich also, wie gewünscht, $v(C_{m-1}) = 1$ gdw $v(A) = \frac{m-2}{m-1}$.

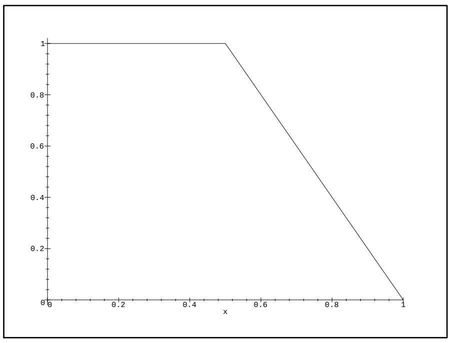
∎



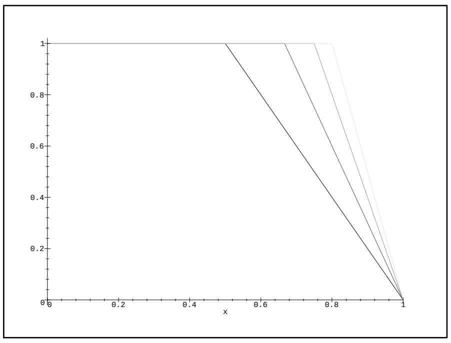
(a) $A \sqcup A \Rightarrow A$



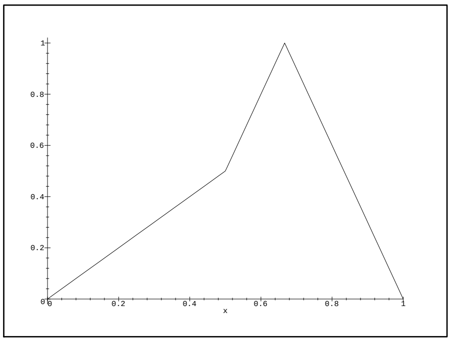
(b) $\bigsqcup^4 A \Rightarrow \bigsqcup^3 A$



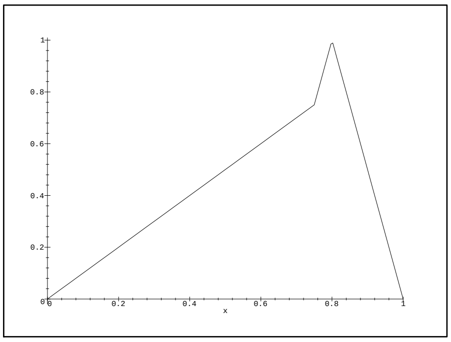
(c) $\neg A \sqcup \neg A$



(d) $\bigsqcup_{i=1}^{m-1} \neg A$ für $m = 3, 4, 5, 6$



(e) C_3



(f) C_5

Abbildung 2.35: Beispiele zu den Funktionen aus Lemma 21

Lemma 22 *Für $n, m \in I\!N$ gilt:*

$$Taut(\mathcal{L}^n_{Łukas}) \subseteq Taut(\mathcal{L}^m_{Łukas})$$
$$gdw$$
$$(m-1) \textit{ ist ein Teiler von } (n-1)$$

Beweis:
Wegen Lemma 19 (1) ist nur noch die Richtung von links nach rechts (von oben nach unten) zu beweisen. Gelte also $Taut(\mathcal{L}^n_{Łukas}) \subseteq Taut(\mathcal{L}^m_{Łukas})$. Wir überlegen uns zunächst, daß dann $W_m \subseteq W_n$ gelten muß. Angenommen, das ist nicht der Fall. Dann gilt $\frac{1}{m-1} \notin W_n$, denn liegt $\frac{1}{m-1}$ in W_n, dann liegen auch alle anderen Elemente aus W_m in W_n. Da W_n abgeschlossen ist unter der Funktion $f(x) = 1 - x$ gilt auch $\frac{m-2}{m-1} \notin W_n$.
Für $F = \bigsqcup^{n-1} \neg C_{m-1}$, wobei C_{m-1} die Formel aus Lemma 21 (4) ist, gilt dann nach Lemma 21 (3) (4) für jede Belegung $v : AL_{Var} \to W_n$:

$$v(F) = \begin{cases} 0 & \text{falls } v(A) = \frac{m-2}{m-1} \\ 1 & \text{sonst} \end{cases}$$

Wegen $\frac{m-2}{m-1} \notin W_n$ folgt also

$$F \in Taut(\mathcal{L}^n_{Łukas})$$

Für $v(A) = \frac{m-2}{m-1}$ ergibt sich $v(F) = 0$ und damit $F \notin Taut(\mathcal{L}^m_{Łukas})$ im Widerspruch zur Voraussetzung. Damit ist $W_m \subseteq W_n$ gezeigt. Inbesondere folgt daraus $\frac{1}{m-1} \in W_n$, d.h. $\frac{1}{m-1} = \frac{k}{n-1}$ für ein geeignetes k. Hieraus folgt $n - 1 = k \cdot (m-1)$ und $(m-1)$ ist in der Tat ein Teiler von $(n-1)$.

∎

Lemma 23 *Sei $1 < k < n$, $3 < n$ und k kein Teiler von $n-1$. Dann ist die folgende Formel eine Tautologie in $\mathcal{L}^n_{Łukas}$*

$$\bigsqcup^{n-1} (\bigsqcap^k \neg A \sqcup (A \sqcap \bigsqcup^{k-1} A))$$

Satz 24 (Satz von McNaughton)
Eine reell-wertige Funktion $f : [0,1]^k \to [0,1]$ ist durch eine $\mathcal{L}^{\aleph_1}_{Łukas}$-Formel definierbar, genau dann, wenn sie die folgenden beiden Bedingungen erfüllt:

1. *$f(x_1, \ldots, x_k)$ ist stetig,*

2. *es gibt endlich viele lineare Polynome $p_1, \ldots, p_s$ von der Form*

$$p_j = b_j + c_{j,1}x_1 + \ldots + c_{j,k}x_k$$

mit Koeffizienten $b_j \in \mathbb{Z}$ und $c_1, \ldots, c_{j,k} \in \mathbb{Z}$, so daß für jedes k-Tupel $\langle r_1, \ldots, r_k \rangle \in [0, 1]^k$ ein $1 \leq j \leq s$ existiert mit

$$f(r_1, \ldots, r_k) = p_j(r_1, \ldots, r_k)$$

Beweis:
Der Beweis ist umfangreich und tiefliegend. Die Orginalarbeit [McNaughton, 1951] enthält einen reinen Existenzbeweise. Ein konstruktiver Beweis wurde von D. Munidici in [Mundici, 1994] gegeben.

■

2.5.2 Unscharfe Logiken

Zum praktischen, ja sogar industriellen Einsatz kamen mehrwertige Logiken in der Form von Logiken mit dem Intervall der rationalen Zahlen zwischen 0 und 1 als Wahrheitswerten. Man nennt diese Logiken *unscharfe Logik* (*fuzzy logic*) oder auch die Theorie der unscharfen Mengen (fuzzy set theory). Es handelt sich dabei um ganze Familien von Logiken, die je nach der Art der Anwendung ausgewählt werden. Ein großer Teil der theoretischen Untersuchungen versucht allgemeine Konstruktionsverfahren für unscharfe Logiken bereitzustellen und allgemeine Tatsachen herzuleiten, die für **alle** nach demselben Konstruktionsverfahren hergestellten Logiken gelten. Wir orientieren uns in der folgenden kurzen Einführung an dem Buch [Gottwald, 1993]. Eine rein verbandstheoretische Darstellung findet man in Kapitel 8 von [Bolc & Borowik, 1992]. Die aktuellste Darstellung der logischen Grundlagen der unscharfen Logik (fuzzy logic) ist [Hájek, 1998].

Wir kommen wieder zurück auf die kurze Diskussion in Abschnitt 2.1.3 darüber, welches die *richtige* Definition einer aussagenlogischen Konjunktion in einer mehrwertigen Logik sein sollte. Hier lassen wir jetzt auch Logiken mit Wahrheitswerten aus dem reellen Intervall $[0, 1]$ zu. Wann würden wir eine Funktion $f : [0, 1]^2 \to [0, 1]$ eine Konjunktion, eine und-Verknüpfung, nennen? Wir stellen hier einen axiomatischen Ansatz zur Beantwortung dieser Frage vor, der in den letzten Jahren zunehmend an Beliebtheit gewonnen hat. Wir beginnen mit einigen, sehr einfachen Forderungen, die unstrittig sind. Eine binäre Funktion, die diesen Minimalanforderungen genügt nennen wir eine t-Norm.

Definition 29 (t-Norm)
Eine zweistellige Funktion $f : [0,1]^2 \rightarrow [0,1]$ heißt eine **t-Norm** *(abkürzend für* triangular norm*) wenn sie die folgenden Eigenschaften erfüllt:*

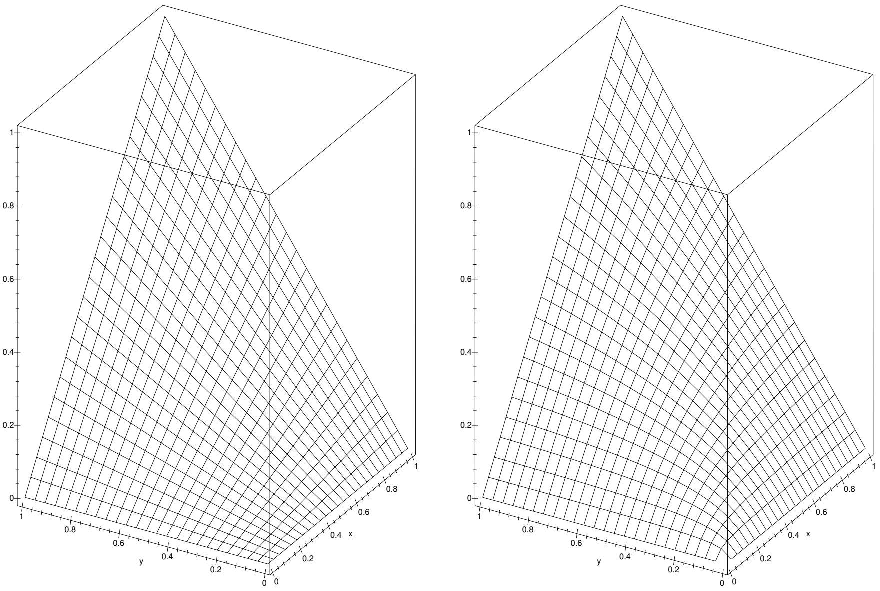
1. *f ist assoziativ und kommutativ,*
2. *für alle $0 \leq A \leq B \leq 1$ und jedes $0 \leq C \leq 1$ gilt $f(A,C) \leq f(B,C)$,*
3. *für jedes $0 \leq C \leq 1$ gilt $f(C,1) = C$*

Es gibt drei Beispiele für t-Normen, die eine prominent wichtige Rolle spielen: die Gödelsche t-Norm, f_G, die Łukasiewicz t-Norm, f_L und die Produkt t-Norm, f_P. Diese sind zusammen mit drei weiteren komplizierteren Beispielen in Abbildung 2.36 zusammengestellt, siehe auch Abbildung 2.36. Diese Liste stammt aus [Gottwald, 1993].

t-Norm	Formel	Parameter
$f_G(A,B)$	$min\{A,B\}$	keine
$f_L(A,B)$	$max\{0, A+B-1\}$	keine
$f_P(A,B)$	$A \cdot B$	keine
$f_D(A,B)$	$\begin{cases} min\{A,B\} & \text{für } A=1 \text{ oder } B=1 \\ 0 & \text{sonst} \end{cases}$	keine
$f^H_\gamma(A,B)$	$\frac{AB}{\gamma+(1-\gamma)\cdot(A+B-AB)}$	$\gamma \geq 0$
r$f^Y_p(A,B)$	$1 - min\{((1-A)^p + (1-B)^p)^{\frac{1}{p}}, 1\}$	$p > 0$

Abbildung 2.36: Beispiele für t-Normen

Es gibt eine einfache, aber vielseitig einsetzbare Methode, aus einer t-Norm weitere zu erhalten.



(a) Produktnorm (b) Die t-Norm f_γ^H für $\gamma = 1$.

Abbildung 2.37: Beispiel zu t-Normen

Lemma 25
Sei $f_1 : [0,1]^2 \to [0,1]$ *eine t-Norm und* $h : [0,1] \to [0,1]$ *eine bijektive, monotone Abbildung, dann ist auch*

$$f_2(x,y) = h^{-1}(f_1(h(x), h(y)))$$

eine t-Norm.

Beweis: Einfaches Nachrechnen.

■

Beispiel 10 *Die Funktion* $k(x,y) = 2^{1-\frac{1}{x}}$ *ist eine bijektive und monotone Funktion von* $[0,1]$ *auf* $[0,1]$ *mit der Umkehrfunktion* $k^{-1}(x,y) = \frac{1}{1-log_2(x)}$. *Also ist*

$$f_2(x,y) = k^{-1}(k(x) * k(y))$$

eine t-Norm. Man rechnet leicht nach, daß f_2 mit der t-Norm f_γ^H aus Abbildung 2.36 für $\gamma = 1$ übereinstimmt.

Definition 30 (Linksseitige Stetigkeit)
Eine t-Norm $f(A, B)$ heißt **linksstetig**, *wenn für alle $A, B \in [0, 1]$ und jede Folge A_i mit $0 \leq A_i \leq A$ und $\lim_{i\to\infty} A_i = A$ gilt*

$$\lim_{i\to\infty} f(A_i, B) = f(A, B)$$

Bis auf f_D sind alle t-Normen in dieser Tabelle linksstetig.

Zu jeder linksstetigen t-Norm f definieren wir eine unscharfe Logik $\mathcal{F}_f$ mit den aussagenlogischen Junktoren $\wedge_f, \vee_f, \rightarrow_f, \neg_f, \leftrightarrow_f$.

$$\begin{array}{lcl} A \wedge_f B & = & f(A, B) \\ A \vee_f B & = & 1 - f(1 - A, 1 - B) \\ A \rightarrow_f B & = & max\{C \mid f(A, C) \leq B\} \\ \neg_f A & = & A \rightarrow_f 0 \\ A \leftrightarrow_f B & = & (A \rightarrow_f B) \wedge_f (B \rightarrow_f A) \end{array}$$

Die geforderte Linksstetigkeit sichert die Existenz eines Maximums für jede Menge $\{C \mid f(A, C) \leq B\}$.
Eine Formel H der Logik $\mathcal{F}_f$ heißt eine Tautologie, wenn die ihr zugeordnete Funktion konstant den Wert 1 hat.

Lemma 26 *Sei f eine linksstetige t-Norm, dann gilt*

1. $A \rightarrow_f B = 1$ *gdw* $A \leq B$
2. $f(A, B) = 1$ *gdw* $A = B = 1$
3. $A \leftrightarrow_f B = 1$ *gdw* $A = B$

Beweis:
(1) Nach Definition folgt aus der linken Seite $max\{C \mid f(A, C) \leq B\} = 1$, also insbesondere $f(A, 1) \leq B$. Da f eine t-Norm ist gilt $f(A, 1) = A$ und damit $A \leq B$. Gilt umgekehrt $A \leq B$, so folgt aus der Monotonie von f auch $f(A, 1) \leq f(B, 1) = B$ und damit $max\{C \mid f(A, C) \leq B\} = 1$.
(2) $f(1, 1) = 1$ ist offensichtlich. Gelte also $f(A, B) = 1$. Aus $A \leq 1$

folgt $1 = f(A, B) \leq f(1, B) = B \leq 1$, also $B = 1$. Dual zeigt man $A = 1$.
(3) Nach Definition von $\leftrightarrow_f$ gilt $A \leftrightarrow_f B = 1$ genau dann, wenn $f(A \rightarrow_f B, B \rightarrow_f A) = 1$. Das ist nach (2) gleichbedeutend mit $A \rightarrow_f B = B \rightarrow_f A = 1$. Mit Hilfe von (1) folgt daraus $A \leq B \leq A$.

■

Lemma 27 *Für jede linksstetige t-Norm f sind die folgenden Formeln Tautologien:*

1. $(A \rightarrow_f B) \rightarrow_f ((A \wedge_f C) \rightarrow_f (B \wedge_f C))$
2. $(A \rightarrow_f B) \rightarrow_f ((C \rightarrow_f A) \rightarrow_f (C \rightarrow_f B))$
3. $(A \rightarrow_f B) \rightarrow_f ((B \rightarrow_f C) \rightarrow_f (A \rightarrow_f C))$

Beweis:
(1) Nach Lemma 26 und der Definition von $\wedge_f$ müssen wir

$$max\{D \mid f(A, D) \leq B\} \leq max\{D \mid f(f(A, C), D) \leq f(B, C)\}$$

zeigen. Dazu genügt es nachzuweisen, daß für jedes D mit $f(A, D) \leq B$ auch $f(f(A, C), D) \leq f(B, C)$ gilt. Aber das ist wegen der Monotonie und Assoziativität von f offensichtlich. Der Leser beweise (2) und (3) als Übungsaufgabe.

■

Lemma 28 *Für jede linksstetige t-Norm f sind die beiden folgenden Formeln Tautologien:*

1. $A \rightarrow_f \neg_f \neg_f A$
2. $\neg_f A \rightarrow_f \neg_f \neg_f \neg_f A$

Beweis:

2.5.3 Übungsaufgaben

Übungsaufgabe 2.5.1 *Zeigen Sie, daß in jeder Łukasiewicz Logik* $\mathcal{L}^n_{Łukas}$ *die Operatoren* $\vee,\wedge$ *und* $\equiv$ *durch die verbleibenden* $\neg$, $\Rightarrow$ *definiert werden können.*
Zeigen Sie, daß auch $\{\neg, \sqcup\}$ *eine funktionale Basis für die Łukasiewicz Logik ist.*

Übungsaufgabe 2.5.2 *Beweisen Sie Lemma 23 für den Spezialfall* $k = 2$.

Übungsaufgabe 2.5.3 *In [Urquhart, 1986] wird als Menge der Wahrheitswerte für die n-wertige Łukasiewicz die Menge* $W^u_n = \{0, 1, \ldots, n-1\} \subset \mathbb{N}$ *benutzt. Außerdem ist* 0 *der höchste und damit einzige ausgezeichnete Wahrheitswert. Die folgende Abbildung stellt eine ordnungserhaltende Isomorphie* $\phi : W^u_n \to W_n$ *her:*

$$\begin{aligned} \phi(k) &= \tfrac{n-1-k}{n-1} \\ \phi^{-1}(k) &= n-1-k\cdot(n-1) \end{aligned}$$

Das Analogon der Implikation $\Rightarrow$ *in* $\mathcal{L}^n_{Łukas}$ *in Urquarts Darstellung bezeichnen wir mit* $\Rightarrow^u$. *Entsprechend das Analogon der Äquivalenx* $\equiv$ *mit* $\equiv^u$. *Wobei gilt*

$$\begin{aligned} p \Rightarrow^u q &= \phi^{-1}(\phi(p) \Rightarrow \phi(q)) \\ p \equiv^u q &= \phi^{-1}(\phi(p) \equiv \phi(q)) \end{aligned}$$

Berechnen Sie $p \Rightarrow^u q$ *und* $p \equiv^u q$.

Übungsaufgabe 2.5.4 *Bestimmen Sie die Funktion* $f^m(A)$, *die von der Formel* $\bigsqcup^m A \Rightarrow \bigsqcup^{m-1} A$ *auf dem rellen Intervall* $[0, 1]$ *erzeugt wird.*

Übungsaufgabe 2.5.5 *Bestimmen Sie die Funktion* $g^m(A)$, *die von der Formel* $C_m = (\bigsqcup^m_{i=1} \neg A) \wedge (A \sqcup \bigsqcap^{m-1}_{i=1} A)$ *auf dem rellen Intervall* $[0, 1]$ *erzeugt wird.*

Übungsaufgabe 2.5.6 *Betrachten sie*

$$Taut(\mathcal{L}^{\aleph_0}_{Łukas}) = \bigcap\{Taut(\mathcal{L}^n_{Łukas}) \mid n \in I\}$$

In Lemma 19 wurde gezeigt, daß diese Gleichung richtig ist für $I = \{n \mid n \geq 2, n \in \mathbb{N}\}$*. Man sieht leicht, daß der dort gegebene Beweis aus funktioniert für alle* $I_k = \{n \mid n \geq k, n \in \mathbb{N}\}$*. Zeigen Sie, daß die angegebene Gleichung für jede unendliche Teilmenge* I *von* $\mathbb{N}$ *gilt.*

Übungsaufgabe 2.5.7 *Zeigen Sie, daß für jede t-Norm* $f : [0,1]^2 \to [0,1]$ *und alle für alle* $A \in [0,1]$ *gilt:*

1. $f(A, 0) = 0$
2. $f(A, A) \leq A$

Übungsaufgabe 2.5.8 *Sei* $f(A, B)$ *die t-Norm* $f^H_\gamma(A, B)$ *aus Abbildung 2.36 für* $\gamma = 0$*. Wir betrachten* $F(A) = f(A, A)$*. Mit* $F^k(A)$ *bezeichnen wir, wie üblich, die k-fache Hintereinanderausführung von* F*, d.h.* $F^k(A) = F(F \ldots (A) \ldots)$*.*

Zeigen Sie, daß

$$F^k(A) = \frac{A}{2^k - (2^k - 1)A}$$

gilt

2.6 Reduktion auf zweiwertige Logik

Als letzten Punkt in diesem Abschnitt wollen wir eine Methode angeben, die es erlaubt $\mathcal{L}_3$-Formeln auf zweiwertige Formeln zu reduzieren. Als Vorbereitung benötigen wir den Begriff der starken Negationsnormalform.

Definition 31 *Eine $\mathcal{L}_3$-Formel A heißt eine* **starke Negationsnormalform**, *wenn in A das starke Negationszeichen $\neg$ höchstens vor atomaren Formeln vorkommt.*
Das schwache Negationszeichen unterliegt keiner Einschränkung.

Lemma 29 *Zu jeder $\mathcal{L}_3$-Formel A gibt es eine $\mathcal{L}_3$-Formel A^{snf} in Negationsnormalform, so daß A und A^{snf} dieselbe dreiwertige Funktion repräsentieren.*

Beweis: Die gesuchte Formel A^{snf} wird induktiv durch die folgenden Regeln aufgebaut:

$$
\begin{array}{lcl}
A^{snf} & = & A \text{ falls } A \text{ atomar ist} \\
(\neg A)^{snf} & = & \neg A \text{ falls } A \text{ atomar ist} \\
(\neg\neg A)^{snf} & = & A^{snf} \\
(\neg \sim A)^{snf} & = & \sim\sim A^{snf} \\
(\neg(A \wedge B))^{snf} & = & (\neg A)^{snf} \vee (\neg B)^{snf} \\
(\neg(A \vee B))^{snf} & = & (\neg A)^{snf} \wedge (\neg B)^{snf} \\
(\neg(\exists x A))^{snf} & = & \forall x(\neg A)^{snf} \\
(\neg(\forall x A))^{snf} & = & \exists x(\neg A)^{snf} \\
(A \wedge B)^{snf} & = & A^{snf} \wedge B^{snf} \\
(A \vee B)^{snf} & = & A^{snf} \vee B^{snf} \\
(\exists x A)^{snf} & = & \exists x A^{snf} \\
(\forall x A)^{snf} & = & \forall x A^{snf} \\
(\sim A)^{snf} & = & \sim A^{snf}
\end{array}
$$

Der behauptete Zusammenhang zwischen A und A^{snf} folgt nun unmittelbar aus Lemma 1.

Die Definition von $(\neg \sim A)^{snf}$ führt ein neues Vorkommen des schwachen Negationszeichens ein. Das ist auf der einen Seite möglich - die angestrebte Normalform macht keine Einschränkungen an die schwache Negation - auf der anderen Seite ist es notwendig, denn die Formel $\neg \sim AidA$ wäre keine Tautologie. Es gilt nur die schwache Äquivalenz $\neg \sim A \equiv A$.

■

Die Transformation von $\mathcal{L}_3$-Formeln in zweiwertige L-Formeln ist nur möglich auf Kosten einer Änderung des Vokabulars, genauer wird nämlich eine $\mathcal{L}_3$-Formel A über dem Vokabular V transformiert in eine $\mathcal{L}$-Formel A_0 über einem Vokabular V_0. Die Konstanten- und Funktionszeichen sind dieselben in V und V_0. Ist p ein n-stelliges Prädikatszeichen in V so gibt es in V_0 statt dessen zwei n-stellige Prädikatszeichen p^+ und p^-.

Definition 32 *Ist A eine Formel in $Fml_3(V)$, so wird die Formel A_0 in $Fml(V_0)$ erhalten, indem man A zuerst nach den Regeln aus Lemma 29 in eine schwach äquivalente starke Negationsnormalform A^{snf} umformt. Anschließend ersetzt man in A^{snf} jede Teilformel der Form $\neg p(t_1, \ldots, t_n)$ durch $p^-(t_1, \ldots, t_n)$ und jede nicht stark negiert vorkommende, atomare Teilformel $p(t_1, \ldots, t_n)$ durch $p^+(t_1, \ldots, t_n)$.*

In A_0 kommt das starke Negationszeichen $\neg$ nicht mehr vor. Identifizieren wir das in A_0 eventuell vorkommende schwache Negationszeichen $\sim$ mit dem einzigen Negationszeichen in der zweiwertigen Sprache $\mathcal{L}$, so liegt A_0 in $Fml(V_0)$.

Beispiel 11

Für $A = p(x) \leftrightarrow q(x)$ erhält man
$$A_0 = (p^-(x) \vee q^+(x)) \wedge (p^+(x) \vee q^-(x))$$

Um den semantischen Zusammenhang zwischen den Formeln A und A_0 genau zu beschrieben müssen wir etwas weiter ausholen, schließlich ist A eine $\mathcal{L}_3$-Formel während A_0 eine zweiwertige Formel ist und noch in einem anderen Vokabular. Auf der rein technischen Ebene besteht die Schwierigkeit darin, $\mathcal{L}_3$-Strukturen $\mathcal{M}$ über V in zweiwertige Strukturen $\mathcal{M}_0$ über dem Vokabular V_0 zu transformieren.

Definition 33
Die Interpretation der Konstanten- und Funktionszeichen ist dieselbe in $\mathcal{M}$und $\mathcal{M}_0$.
Ist p ein n-stelliges Prädikatszeichen in V, so setzen wir:

$$\begin{array}{lll} \mathcal{M}_0 \models p^+(m_1,\ldots,m_n) & gdw & v_{\mathcal{M}}(p(m_1,\ldots,m_n)) = 1 \\ \mathcal{M}_0 \models p^-(m_1,\ldots,m_n) & gdw & v_{\mathcal{M}}(p(m_1,\ldots,m_n)) = 0 \end{array}$$

Lemma 30 *Für jede $\mathcal{L}_3$-Struktur $\mathcal{M}$ über V, jede Formel $A(x_1,\ldots,x_k)$ aus $Fml_3(V)$ mit den freien Variablen $x_1,\ldots,x_k$ und jedes k-Tupel $m_1,\ldots$ $\ldots,m_k$ von Elementen aus $\mathcal{M}$ gilt:*

$$v_{\mathcal{M}}(A(m_1,\ldots,m_k)) = 1 \textit{ gdw } \mathcal{M}_0 \models A_0(m_1,\ldots,m_k)$$

Beweis: Man kann zunächst ohne Einbuße an Allgemeinheit annehmen, daß A eine starke Negationsnormalform ist. Der Beweis wird dann durch einfache Induktion über die Komplexität von A geführt.

Die Abbildung von $\mathcal{L}_3$-Strukturen über V in $\mathcal{L}$-Strukturen über V_0 ist nicht surjektiv, d.h. es gibt $\mathcal{L}$-Strukturen $\mathcal{N}$ über dem Vokabular V_0, so daß für keine $\mathcal{L}_3$-Struktur $\mathcal{M}$ über V die Gleichheit $\mathcal{N} = \mathcal{M}_0$ gilt. Den genauen Sachverhalt beschreibt das folgende Lemma:

Lemma 31 *Zu einer zweiwertigen Struktur $\mathcal{N}$ über dem Vokabular V_0 gibt es eine $\mathcal{L}_3$-Struktur $\mathcal{M}$ über V mit $\mathcal{M}_0 = \mathcal{N}$ genau dann, wenn für jedes n-stellige Prädikatszeichen $p \in V$ gilt*

$$\mathcal{N} \models \forall x_1 \ldots \forall x_n \sim (p^+(x_1,\ldots,x_n) \wedge p^-(x_1,\ldots,x_n))$$

Beweis: Einfach.

Satz 32 *Sei Σ eine Menge von $Fml_3(V)$-Formeln und A eine einzelne $Fml_3(V)$-Formel, sei weiter $\Sigma_0 = \{B_0 : B \in \Sigma\}$ und T die Menge aller Formeln der Form*

$\forall x_1 \ldots \forall x_n \sim (p^+(x_1,\ldots,x_n) \wedge p^-(x_1,\ldots,x_n))$

für jedes n und jedes n-stellige Prädikatszeichen $p \in V$. Dann gilt:

1. *$\Sigma \vdash_3 A$ gdw $\Sigma_0 \cup T \vdash A_0$*
2. *Insbesondere gilt also:*
   A ist eine (dreiwertige) Tautologie gdw $T \vdash A_0$.

Beweis: Gelte zunächst $\Sigma \vdash_3 A$. Zu jedem V_0-Modell von $\Sigma_0 \cup T$ gibt es nach Lemma 31 eine dreiwertige V-Struktur $\mathcal{M}$ mit $\mathcal{M}_0 = \mathcal{N}$. Nach Lemma 30 gilt $v_{\mathcal{M}}(B) = 1$ für alle $B \in \Sigma$, woraus nach Voraussetzung $v_{\mathcal{M}}(A) = 1$ folgt. Lemma 30 liefert jetzt wieder $\mathcal{M}_0 \models A_0$, i.e. $\mathcal{N} \models A_0$. Damit ist $\Sigma_0 \cup T \vdash A_0$ gezeigt.
Gelte jetzt umgekehrt $\Sigma_0 \cup T \vdash A_0$. Ist $\mathcal{M}$ eine dreiwertige V-Struktur, so daß $v_{\mathcal{M}}(B) = 1$ für alle $B \in \Sigma$, so gilt nach Lemma 30 $\mathcal{M}_0 \models \Sigma_0$. Nach Voraussetzung haben wir dann auch $\mathcal{M}_0 \models A_0$, woraus wieder mit Lemma 30 $v_{\mathcal{M}}(A) = 1$ folgt. Insgesamt ist damit $\Sigma \vdash_3 A$ gezeigt.

2.6.1 Übungsaufgaben

Übungsaufgabe 2.6.1 *Zeigen Sie, daß zu jeder $Fml(V_0)$-Formel B eine $Fml_3(V)$-Formel A existiert mit $A_0 = B$.*

Übungsaufgabe 2.6.2 *Für eine $\mathcal{L}_3$-Struktur $\mathcal{M}$ sei $\mathcal{M}_0$ konstruiert wie in Definition 33 beschrieben.*

Finden Sie für jede Formel $A(x_1, \ldots, x_k)$ aus $Fml_3(V)$ mit den freien Variablen $x_1, \ldots, x_k$ eine Formel A_u, so daß für jedes k-Tupel $m_1, \ldots \ldots, m_k$ von Elementen aus $\mathcal{M}$ gilt:

$$v_{\mathcal{M}}(A(m_1, \ldots, m_k)) = u \text{ gdw } \mathcal{M}_0 \models A_u(m_1, \ldots, m_k)$$

Übungsaufgabe 2.6.3 *Geben Sie zu den folgenden $\mathcal{L}_3$-Formeln ihre $\mathcal{L}_2$-Übersetzungen an:*

1. $\forall x(p(x) \vee \neg p(x))$
2. $\forall x(p(x) \vee \sim p(x))$
3. $\forall x(\neg p(x) \vee \sim \neg p(x))$
4. $\forall x(\sim p(x) \vee \neg \sim p(x))$

2.7 Abstrakte Konsequenzrelationen

In den vorangegangenen Kapiteln haben wir einige konkret definierte Konsequenzrelationen kennengelernt. Ist z. B. Σ eine Menge von Ł3-Formeln und φ eine Ł3-Formel ohne freie Variablen, so war in Definition 12 auf Seite 27 die Relation $\Sigma \vdash \varphi$ dadurch definiert, daß für jede Ł3-Struktur $(\mathcal{M}, v)$, in der alle Formeln aus Σ den Wahrheitswert 1 haben, auch φ den Wahrheitswert 1 hat. Ein zweites Beispiel wird gegeben durch die aussagenlogischen Formeln über einer Operatorenmenge F und einer zugehörigen F-Algebra $\mathcal{M}$. Ist Σ eine Menge aussagenlogischer Formeln, A eine aussagenlogische Formel, so ist A eine $\mathcal{M}$-Konsequenz aus Σ, in Zeichen $\Sigma \vdash_{\mathcal{M}} A$, falls für jede Belegung v der Aussagenvariablen mit Wahrheitswerten (=Elementen des Universums von $\mathcal{M}$), so daß für alle $B \in \Sigma$ $v(B)$ ein designierter Wahrheitswert ist, auch $v(A)$ ein designierter Wahrheitswert ist. Die Vielzahl der betrachteten nichtklassischen Logiken mit den jeweiligen Konsequenzrelationen hat dazu geführt, den Begriff einer Konsequenzrelation axiomatisch zu fassen, [Tarski, 1930], [Tarski, 1935]. Siehe auch [Gabbay, 1994].

Sei ALF (falls erforderlich schreiben wir genauer ALF(F)) die Menge aller aussagenlogischen Formeln über der Operatorenmenge F. Eine zweistellige Relation $\vdash$ zwischen Teilmengen Σ von ALF und Elementen A aus ALF, $\Sigma \vdash A$, heißt eine **Konsequenzrelation** (manchmal sagen wir zur Betonung des axiomatischen Zugangs auch abstrakte Konsequenzrelation), falls die folgenden Axiome erfüllt sind:

Axiom I: Für jedes $A \in \Sigma$ gilt $\Sigma \vdash A$.

Axiom II: Gilt $\Sigma_0 \vdash A$ und $\Sigma_0 \subseteq \Sigma_1$, dann gilt auch $\Sigma_1 \vdash A$. (Monotonie)

Axiom III: Gilt $\Delta \vdash A$, und für jedes $B \in \Delta$ gilt auch $\Sigma \vdash B$, dann gilt $\Sigma \vdash A$.

Eine Konsequenzrelation $\vdash$ besitzt die **Endlichkeitseigenschaft**, falls gilt

Axiom IV: Wenn $\Sigma \vdash A$, dann gibt es eine endliche Teilmenge $\Sigma_0 \subseteq \Sigma$ mit $\Sigma_0 \vdash A$.

Ist A eine aussagenlogische Formel und s eine Abbildung, die jeder Aussagenvariablen p eine aussagenlogische Formel zuordnet, so wird mit $s(A)$ das

Ergebnis bezeichnet, das entsteht, wenn jede Aussagenvariable p in A durch $s(p)$ ersetzt wird.

Eine Konsequenzrelation $\vdash$ heißt **strukturell**, wenn gilt

Axiom V: Gilt $\Sigma \vdash A$, dann gilt für jede Substitution s auch $s(\Sigma) \vdash s(A)$.
(Hierbei steht natürlich $s(\Sigma)$ für die Menge $\{s(B) : B \in \Sigma\}$.)

Gilt für alle $A \in$ ALF die Beziehung $\Sigma \vdash A$, dann heißt Σ **widersprüchlich**.

Eine Konsequenzrelation $\vdash$ heißt **uniform**, falls gilt:
$\{B\}$ sei als nicht widersprüchlich vorausgesetzt, und
B habe keine Aussagenvariable mit Σ oder A gemeinsam,
dann folgt aus

$$\Sigma \cup \{B\} \vdash A$$

auch

$$\Sigma \vdash A.$$

Lemma 33 *Sei $\mathcal{M}$ eine endliche F-Algebra, dann ist $\vdash_{\mathcal{M}}$ eine uniforme, strukturelle Konsequenzrelation mit Endlichkeitseigenschaft.*

Beweis:
Einzig die Endlichkeitseigenschaft ist nicht trivial. Sei das Universum von $\mathcal{M}$ die Menge $\{0, \ldots, r-1\}$. Gelte $\Sigma \vdash_{\mathcal{M}} A$, aber für alle endlichen Teilmengen $\Sigma_0 \subseteq \Sigma$ gelte nicht $\Sigma_0 \vdash_{\mathcal{M}} A$. Sei $P = \{p_1, p_2, \ldots\}$ die Menge aller Aussagenvariablen in $\Sigma \cup \{A\}$. Wir nennen eine Belegung $w : P \to \mathcal{M}$ ein Modell für $\Sigma_0 \cup \{\neg A\}$, falls für alle $B \in \Sigma_0$ $w(B)$ ein designierter Wahrheitswert von $\mathcal{M}$ ist, $w(A)$ aber nicht. Wir wollen schrittweise, durch Induktion nach n, eine Belegung $w : \{p_1, \ldots, p_n\} \to \mathcal{M}$ konstruieren, so daß für jede endliche Teilmenge $\Sigma_0 \subseteq \Sigma$ ein Modell v von $\Sigma_0 \cup \{\neg A\}$ existiert mit $v(p_1) = w(p_1), \ldots, v(p_n) = w(p_n)$.

Wir führen den Induktionsschritt von n nach $n+1$ im Detail vor.

Der Induktionsanfang wird durch eine einfache Modifikation dieses Arguments bewiesen.

Sei V_n die Menge aller Belegungen v, so daß

$$v(p_1) = w(p_1), \ldots, v(p_n) = w(p_n).$$

Falls für ein $j, 0 < j < r$, für jede endliche Teilmenge $\Sigma_0 \subseteq \Sigma$ ein Modell v von $\Sigma_0 \cup \{\neg A\}$ aus V_n existiert mit $v(p_{n+1}) = j$, dann setzen wir $w(p_{n+1}) = j$ und sind fertig. Ist das nicht der Fall, so gibt es für jedes $j, 0 < j < r$, eine endliche Teilmenge Σ_j von Σ, so daß für jedes Modell v von $\Sigma_j \cup \{\neg A\}$ aus V_n gelten muß $v(p_{n+1}) \neq j$.

Wir setzen $w(p_{n+1}) = 0$ und müssen noch nachweisen, daß damit die gewünschte Eigenschaft erfüllt ist. Sei also Σ_0 eine endliche Teilmenge von Σ. Dann ist $\Delta = \Sigma_0 \cup \Sigma_1 \cup \ldots \cup \Sigma_{r-1}$ ebenfalls eine endliche Teilmenge von Σ, die nach Induktionsvoraussetzung ein Modell $v \in V_n$ besitzt. Nach Wahl von $\Sigma_j, 0 < j < r$, muß dann $v(p_{n+1}) = 0 = w(p_{n+1})$ gelten. Damit ist die Konstruktion von w abgeschlossen.

Für jede Formel B aus Σ ist $w(B)$ ein designierter Wahrheitswert. Das sieht man wie folgt ein: Sei p_n die Aussagenvariable mit dem größten Index in B. Nach Konstruktion gibt es ein Modell $v \in V_n$ für $\{B, \neg A\}$. Da $v(B) = w(B)$ ist, folgt die Behauptung. Ebenso zeigt man, daß $w(A)$ kein designierter Wahrheitswert ist. Das steht im Widerspruch zu $\Sigma \vdash A$.

■

Satz 34 *Sei $\vdash$ eine strukturelle Konsequenzrelation mit Endlichkeitseigenschaft.*

Es gibt eine F-Algebra $\mathcal{M}$ mit $\vdash = \vdash_{\mathcal{M}}$
gdw.
$\vdash$ ist uniform.

Beweis: Dieses Resultat wurde zuerst in [J. Los, 1958] bewiesen. Einen Beweis findet man z.B. auch in [Urquhart, 1986] auf den Seiten 78ff.

Wir skizzieren die Beweisidee. Sei $\vdash$ eine strukturelle Konsequenzrelation mit der Endlichkeitseigenschaft. Wir betrachten alle Paare (Σ, A) wobei $\Sigma \subseteq ALF$ und $A \in ALF$, für die $\Sigma \not\vdash A$ gilt. In einem ersten Beweisschritt konstruiert man zu jedem solchen Paar eine Struktur $\mathcal{M}_{\Sigma,A}$ mit den Eigenschaften

1. $\vdash \;\subseteq\; \vdash_{\mathcal{M}_{\Sigma,A}}$
2. $\Sigma \not\vdash_{\mathcal{M}_{\Sigma,A}} A$
   d.h. es gilt $\mathcal{M}_{\Sigma,A} \models \Sigma$ und $\mathcal{M}_{\Sigma,A} \not\models A$

Im zweiten Schritt des Beweises betrachtet man das direkte Produkt $\mathcal{M}$ alle Strukturen $\mathcal{M}_{\Sigma,A}$ für die Paare (Σ, A) mit $\Sigma \not\vdash A$. Da die von $\mathcal{M}$ induzierte Konsequenzrelation $\vdash_{\mathcal{M}}$ gerade der mengentheoretische Durchschnitt aller $\vdash_{\mathcal{M}_{\Sigma,A}}$ ist, ergibt sich unmittelbar

$$\vdash \; = \; \vdash_{\mathcal{M}_{\Sigma,A}}$$

Interessant sind die Details des ersten Schrittes. Sei also $\vdash$ gegeben und Σ und A fixiert mit $\Sigma \not\vdash A$. Einfachheitshalber bezeichnen wir die zu konstruierende Struktur $\mathcal{M}_{\Sigma,A}$ mit $\mathcal{N}$. Als Universum N von $\mathcal{N}$ benutzen wir die Menge ALF aller Formeln. Die Definition der Funktionszeichen erfolgt, wie in Herbrand-Strukturen durch *lazy evaluation*, i.e. die Interpretation $f^{\mathcal{M}}$ eines, sagen wir zweistelligen, Funktionszeichens f liefert für Argumente B_1,$B_2 \in ALF$ den Wert $f^{\mathcal{M}}(B_1, B_2) = f(B_1, B_2)$. Die Menge D der ausgezeichneten Wahrheitswerte ist gegeben durch die Definition

$$D = \{B \mid \Sigma \vdash B\}$$

Für die so definierte Struktur $\mathcal{N}$ fallen die Begriffe einer Belegungsfunktion und einer Substitution zusammen, beides sind Abbildungen $v : Var \to N$ von der Menge der Variablen in das Universum N von $\mathcal{N}$. Als Konsequenz dieser Beobachtung fallen auch das Ergebnis der Auswertung einer Formel unter einer Belegung v und das Ergebnis der Ausführung der Substitution zusammen. Betrachten man z.B. die identische Substitution $v(p) = p$ für alle aussagenlogischen Variablen p, dann liefert die Auswertung jeder Formel bezüglich v sich selbst als Ergebnis. Jede Formel $B \in \Sigma$ liefert also einen ausgezeichneten Wahrheitswert aber A nicht. (Man beachte, daß man hierzu von Axiom I Gebrauch machen muss.) Damit ist schon

$$\Sigma \not\vdash_{\mathcal{N}} A$$

gezeigt. Um die noch ausstehende Inklusionsbeziehung nachzuweisen betrachten wir Σ_1 und A_1 mit $\Sigma_1 \vdash A_1$ und wollen $\Sigma_1 \vdash_{\mathcal{N}} A_1$ zeigen. Dazu betrachten wir eine beliebige Bewertung $v : Var \to N$, so daß für alle $B \in \Sigma_1$ gilt $v(B) \in D$. Es bleibt zu argumentieren, daß $v(A_1) \in D$ gilt. Als erstes bemerken wir, daß mit $\Sigma_1 \vdash A_1$ auch

$$v(\Sigma_1) \vdash v(A_1) \tag{2.1}$$

gilt, da $\vdash$ strukturell ist. Nach Definition von D gilt außerdem

$$\text{für alle } B \in \Sigma_1 \text{ auch } \Sigma \vdash v(B) \tag{2.2}$$

Aus 2.1 und 2.2 folgt mit der Transitivität (Axiom III) $\Sigma \vdash v(A_1)$ und damit, wie gewünscht $v(A_1) \in D$.

■

Ein weiteres interessantes Einzelresultat über abstrakte Konsequenzrelationen finden man in [Bolc & Borowik, 1992, p. 100f]

2.7.1 Übungsaufgaben

Übungsaufgabe 2.7.1 *Sei $\vdash$ eine Konsequenzrelation.*
Definiere $\vdash_e$ durch:

$$\Sigma \vdash_e A \qquad \textit{gdw.} \qquad \textit{es gibt eine endliche Teilmenge } \Sigma_0 \subseteq \Sigma \textit{ mit } \Sigma_0 \vdash A.$$

Zeigen Sie

1. *$\vdash_e$ ist eine Konsequenzrelation mit der Endlichkeitseigenschaft.*
2. *Für alle endlichen Mengen Σ gilt:*
   $\Sigma \vdash A \qquad$ *gdw.* $\qquad \Sigma \vdash_e A$.
3. *Ist $\vdash$ strukturell (uniform), dann auch $\vdash_e$.*

Übungsaufgabe 2.7.2 *Zeigen Sie, daß Axiom II aus Axiom I und Axiom III folgt.*

Kapitel 3

Modale Logiken

Die modale Logik beschäftigt sich nicht nur mit der Wahrheit oder der Falschheit einer Aussage sondern auch mit den *Modalitäten* des Wahr- bzw. Falschseins. Wir betrachten in diesem Kapitel Logiksysteme, die hervorgegangen sind aus dem Studium der Aussagen, die *notwendigerweise*, bzw. *möglicherweise* wahr sind. Die Philosophen nennen diese Modalitäten *alethisch* im Unterschied zu den *epistemischen* oder *deontischen* Modalitäten, bei denen nach der Art der Zuordnung eines Wahrheitswertes oder nach der Einordnung in ein Normensystem des Erlaubten und Verbotenen differenziert wird. Philosophische Beiträge zur modalen Logik sind schon aus dem Mittelalter, sogar aus der Antike bekannt, mit der unausweichlichen Referenz auf Aristoteles. Als Beginn einer formalen Behandlung der Modallogik wird allgemein die Arbeit von C. I. Lewis angesehen, [Lewis, 1918]. In den letzten zehn, fünfzehn Jahren ist die Modallogik verstärkt ins Blickfeld der Informatik geraten, angeregt zum Teil durch Fragestellungen aus der Künstlichen Intelligenz aber auch durch einen breiteren Trend zu formal-logischen Methoden.

Konnten die Autoren Hughes und Cresswell 1968 in ihrem Buch, [Hughes & Cresswell, 1972], noch behaupten ein Bild der gesamten Modallogik zu dieser Zeit zu geben, so ist heute ein Überblick, der einigermaßen Anspruch auf Vollständigkeit erheben könnte, schlichtweg unmöglich. Die folgende Liste enthält ausgewählte Monographien und Übersichtsartikel, die nicht einem speziellen Thema innerhalb der Modallogik gewidmet sind und keine besonderen Vorkenntnisse voraussetzen.

- Melvin Fittings Kapitel im Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 1, [Fitting, 1993],
- Part One in den CSLI Lecture Notes von Robert Goldblatt, [Goldblatt, 1987],
- das Kapitel von Colin Stirling im zweiten Band des Handbook of Logic in Computer Science, [Sterling, 1992],
- die Kapitel über modale Aussagenlogik von Robert Bull und Krister Segerberg und über modale Quantorenlogik von James Garson in Band II des Handbook of Philosophical Logic, [Bull & Segerberg, 1984], [Garson, 1984],

- die Fortsetzung ihres 68-er Buches von Hughes and Cresswell, [Hughes & Cresswell, 1984],
- das Kapitel von Peter Steinacker in der Einführung in die Nichtklassische Logik des Akademie-Verlags, Berlin, [Steinacker, 1990]

3.1 Einführung in die modale Aussagenlogik

Der folgende Text bietet auf zwei verschiedenen Wegen einen Einstieg in die Modallogik. Der erste Zugang führt über eine axiomatische Begründung des logischen Schließens mit dem Modaloperator „die Person A weiß, daß die Aussage ϕ wahr ist“. Die zweite Möglichkeit beginnt mit der Semantik modaler Logiken. Motiviert durch eine Abstraktion von dem wohlbekannten Begriff der Transitionssysteme gelangt man zu Strukturen, in denen modallogische Formeln interpretiert werden, den sogenannten Kripke-Strukturen.

3.1.1 Eine Logik des Wissens

Das folgende Puzzle ist in vielen Varianten weit verbreitet:

> Drei Weisen werden Hüte aufgesetzt, jedem genau einen. Die Hüte sind entweder weiß oder schwarz, und jedem ist bekannt, daß mindestens ein schwarzer Hut mit dabei ist. Jeder Beteiligte sieht, welche Hüte die anderen beiden aufsitzen haben und soll erschließen, welchen Hut er aufsitzen hat, natürlich ohne in einen Spiegel zu schauen, den Hut abzunehmen oder ähnliches. Nach einer Weile sagt einer der Weisen: „Ich weiß nicht, welchen Hut ich aufhabe.“ Nach einer weiteren Pause des Nachdenkens sagt ein zweiter: „Ich weiß auch nicht, welchen Hut ich aufhabe.“ „Dann“, sagt der Dritte, „weiß ich, daß ich einen schwarzen Hut aufhabe.“

Bevor wir mit der Formalisierung beginnen, wollen wir die Lösung verraten. Wir versetzen uns in die Lage des dritten Weisen, des weisesten Weisen. Er überlegt sich, nachdem sein erster Kollege zugegeben hat, daß er nicht weiß, welchen Hut er trägt, daß in diesem Fall er und sein zweiter Kollege nicht beide weiße Hüte aufhaben können, denn sonst wüßte der erste, daß er den dann einzigen schwarzen Hut trägt. Der dritte Weise ist sich außerdem darüber im Klaren, daß sein zweiter Kollege dieselbe Schlußfolgerung anstellt. Als dieser jedoch auch erklärt, er wisse nicht, welchen Hut er aufhabe, weiß der dritte, daß er keinen weißen Hut aufsitzen hat.

Die Lösung macht deutlich, daß wesentlich logische Schlüsse benutzt werden über das Wissen der Beteiligten. Wichtig ist nicht nur die Tatsache, daß der zweite oder der dritte Weise einen schwarzen Hut aufhaben muß, sondern auch, daß der zweite das weiß und sogar, daß der dritte weiß, daß der zweite diese Tatsache kennt.

Die Formalisierung des Puzzles geschieht völlig im Rahmen der Aussagenlogik. Die auftretenden Aussagenvariablen sollen die folgenden intuitiven Bedeutungen haben:

S_i	der i-te Weise hat einen schwarzen Hut auf
W_i	der i-te Weise hat einen weißen Hut auf

jeweils für $i \in \{1, 2, 3\}$.

Wir benutzen die Formel $\Box_i\ A$ um auszudrücken, daß der i-te Weise weiß, daß die Formel A wahr ist.

Welche logischen Schlußregeln sind wir bereit in einem formalen Beweis zu akzeptieren? Natürlich wollen wir auf die bekannten aussagenlogischen Tautologien und die vertraute modus ponens Regel nicht verzichten, wobei in beiden Fällen auch modallogische Instanzen mit eingeschlossen sind. Die Formel $\Box_1\ A \vee \neg\Box_1\ A$ ist also eine Tautologie, über die Allgemeingültigkeit der Formel $\Box_1\ A \vee \Box_1(\neg A)$ ist dagegen noch nichts ausgesagt. Den Akteuren in unserem Puzzle sind nicht alle Fakten der sie umgebenden Szenerie bekannt, wir können aber annehmen, daß sie alle allgemeingültigen Fakten kennen. Das sind insbesondere alle Tautologien, aber in der vorliegenden Situation auch alle bekanntgemachten Spielregeln, wie z.B., daß es nur schwarze und weiße Hüte gibt und mindestens einen schwarzen. In der nachfolgenden formalen Argumentation erlauben wir daher die Anwendung der Schlußregel (Generalisierungsregel)

$$\frac{A}{\Box A} \tag{G}$$

Wir benutzen die Regel (G) für jede Instanziierung des Modaloperators $\Box$ durch einen der Operatoren $\Box_i$. Wenn ich weiß, daß heute Samstag ist und ich außerdem weiß, daß es jeden Samstag frische Brötchen zum Frühstück gibt, dann weiß ich natürlich auch, daß es heute frische Brötchen zum Frühstück gibt. In einem formalen Beweis wollen wir deshalb auch Instanzen des folgenden Schemas als Axiome zulassen:

$$(\Box A \wedge \Box(A \to B)) \to \Box B \tag{K}$$

Man beachte, daß dieses Axiomenschema aussagenlogisch äquivalent ist zu

$$\Box(A \to B) \to (\Box A \to \Box B) \tag{K}$$

Aus den bisher zugestandenen logischen Axiomen und Schlußregeln folgen abgeleitete Regeln und weitere Axiome, z.B.

$$\frac{A \to B}{\Box A \to \Box B} \tag{I}$$

$$\frac{A \leftrightarrow B}{\Box A \leftrightarrow \Box B} \tag{E}$$

Das ist leicht einzusehen: Aus der Prämisse $A \leftrightarrow B$ folgt mit der Regel (G) $\Box(A \leftrightarrow B)$. Mit Axiom (K) und modus ponens folgt die gewünschte Folgerung der abgeleiteten Regel.

Besonders hilfreich wird für uns das folgende abgeleitete modallogische Axiomenschema sein:

$$\Box A \wedge \Box B \leftrightarrow \Box(A \wedge B) \tag{M}$$

Die logische Herleitung von (M) beginnt mit dem aussagenlogischen Axiom $(A \wedge B) \to A$, Regel (I) liefert $\Box(A \wedge B) \to \Box A$. Analog erhält man $\Box(A \wedge B) \to \Box B$. Aus beiden Formeln folgt allein mit aussagenlogischen Mitteln bereits eine Implikation von (M), nämlich

$$\Box(A \wedge B) \to (\Box A \wedge \Box B)$$

Zum Beweis der umgekehrten Implikation betrachten wir zunächst die Instanz $\Box(B \to (A \to (A \wedge B))) \to (\Box B \to \Box(A \to (A \wedge B))$ von Axiom (K). Da die erste Prämisse $\Box(B \to (A \to (A \wedge B)))$ aus der aussagenlogischen Tautologie $B \to (A \to (A \wedge B))$ mit der Regel (G) ableitbar ist erhalten wir mit modus ponens

$$\Box B \to \Box(A \to (A \wedge B))$$

Die Formel $\Box(A \to (A \wedge B)) \to (\Box A \to \Box(A \wedge B))$ ist ebenfalls eine Instanz des Axioms (K). Unter Ausnutzung der Transitivität der Implikation erhalten wir aus den letzten beiden Formel auch $\Box B \to (\Box A \to \Box(A \wedge B))$ als ableitbare Formel und die ist aussagenlogisch äquivalent zu

$$\Box A \wedge \Box B \to \Box(A \wedge B)$$

Wie schon für die Regel (G) ist auch in der Benutzung der Axiomenschemata (K) und (M) der Operator $\Box$ durch ein $\Box_i$ zu ersetzen. Das sind auch schon alle logischen Axiome und Regeln, die wir benutzen werden. Dazu kommen natürlich noch die Fakten der Ausgangssituation, z.B. $S_1 \vee S_2 \vee S_3$, aber auch $S_1 \to (\Box_2 S_1 \wedge \Box_3 S_1)$, denn wenn der 1-te Weise einen schwarzen Hut aufhat, dann sehen das die anderen beiden und dann wissen sie auch dieses Faktum. Mehr noch: wenn S_1 gilt, dann weiß auch der 3-te Weise, daß der zweite das weiß. Wir haben also als ein Faktum $S_1 \to \Box_3\Box_2 S_1$. Schließlich kommen noch zwei wichtige Fakten hinzu, nämlich die Aussagen der ersten beiden Weisen. Das Eingeständnis ihres Nichtwissens vergrößert das Wissen des dritten.

$$\neg\Box_1 S_1 \qquad (B_1)$$

$$\neg\Box_2 S_2 \qquad (B_2)$$

Es gilt natürlich auch $\neg\Box_1 W_1$ und $\neg\Box_2 W_2$, aber die folgende Argumentation macht von diesen Fakten keinen direkten Gebrauch.

Wir werden in der folgenden formalen Herleitung von Fakten dieser Art freien Gebrauch machen.

$$W_3 \wedge W_2 \to S_1 \tag{3.1}$$

Faktum

$$\Box_1(W_3 \wedge W_2 \to S_1) \tag{3.2}$$

aus 3.1 mit (G)

$$\Box_1(W_3 \wedge W_2 \to S_1) \wedge \neg\Box_1(S_1) \to \neg\Box_1(W_3 \wedge W_2) \tag{3.3}$$

aus (K) und Aussagenlogik

$$\neg\Box_1(W_3 \wedge W_2) \tag{3.4}$$

aus 3.2, 3.3 und Faktum B_1 mit (MP)

$$(W_3 \to \Box_1 W_3) \wedge (W_2 \to \Box_1 W_2) \tag{3.5}$$

Fakten

$$W_3 \wedge W_2 \to \Box_1 W_3 \wedge \Box_1 W_2 \tag{3.6}$$

mit AL aus 3.5

$$W_3 \wedge W_2 \to \Box_1(W_3 \wedge W_2) \tag{3.7}$$

mit (M), AL und 3.6

$$\neg(W_3 \wedge W_2) \tag{3.8}$$

mit AL aus 3.7 und 3.4

$$\Box_2(W_3 \to \neg W_2) \tag{3.9}$$

mit AL und Regel (G) aus 3.8

$$\Box_2 W_3 \to \Box\neg W_2 \tag{3.10}$$

mit Axiom (K) und AL aus 3.9

$$\neg\Box\neg W_2 \to \neg\Box_2 W_3 \tag{3.11}$$

Kontraposition von 3.10

$$\neg W_2 \to S_2 \tag{3.12}$$

Faktum

$$\Box_2(\neg W_2 \to S_2) \tag{3.13}$$

mit Regel (G) aus 3.12

$$\Box_2\neg W_2 \to \Box_2 S_2 \tag{3.14}$$

mit (MP), Axiom (K) aus 3.13

$$\neg\Box_2\neg W_2 \tag{3.15}$$

aus 3.14 mit AL und Faktum B_2

$$\neg\Box_2 W_3 \tag{3.16}$$

aus 3.15 und 3.11

$$\neg W_3 \tag{3.17}$$

aus 3.16, dem Faktum $W_3 \to \Box_2 W_3$ und AL

$$S_3 \tag{3.18}$$

aus 3.17 mit dem Faktum $\neg W_3 \to S_3$

$$\Box_3 S_3 \tag{3.19}$$

mit Regel (G) aus 3.18 Wir fassen die im vorangegangenen Beispiel implizit eingeführten Begriffe zusammen.

Axiomenschemata	alle AL-Tautologien	
	$\Box(A \to B) \to (\Box A \to \Box B)$	(K)
Regeln	$\dfrac{A\ ,\ A \to B}{B}$	(MP)
	$\dfrac{A}{\Box A}$	(G)

Abbildung 3.1: Das System **K**

Definition 34 *Die Menge, Fml_{ALmod}, der Formeln der modalen Aussagenlogik ist die kleinste Menge mit den nachfolgenden Eigenschaften:*

1. *jede aussagenlogische Variable ist ein Element von Fml_{ALmod}.*
2. *für $F_1, F_2 \in Fml_{ALmod}$ liegen auch $F_1 \wedge F_2$, $F_1 \vee F_2$, $F_1 \to F_2$ und $\neg F_1$ in Fml_{ALmod}.*
3. *für $F \in Fml_{ALmod}$ gilt auch $\Box F \in Fml_{ALmod}$ und $\Diamond F \in Fml_{ALmod}$.*

Eine Formel $\Box A$ wird dabei gelesen als „Box A“ , „notwendigerweise A “oder „A ist notwendig “. Die Formel $\Diamond A$ liest man als „Diamant A“ „möglicherweise A“oder „A ist möglich “.

Eine Eigenheit der modalen Logik ist die Vielzahl der verschiedenen Logiksysteme, siehe Abbildung 3.2. Wir werden einige davon genauer kennenlernen, einige nur erwähnen. Das einfachste Logiksystem, genannt **K**, haben wir schon gesehen. Es ist in Abbildung 3.1 noch einmal zusammengefasst.

Definition 35
Jedes modallogische System **S** *induziert eine Ableitbarkeitsrelation, die wir mit $\models_{\mathbf{S}}$ bezeichnen. Wie üblich gilt dabei die Relation $\Sigma \models_{\mathbf{S}} A$ für eine Formelmenge $\Sigma \subseteq Fml_{ALmod}$ und eine Formel $A \in Fml_{ALmod}$, wenn es einen Beweis für A gibt, der nur die Formeln aus Σ und die Axiome und Schlußregeln von* **S** *benutzt.*

3.1.2 Kripke Strukturen

Den zweiten Einstieg in die modale Logik beginnen wir mit dem wohlbekannten Konzept der Transitionssysteme oder endlichen Automaten. Ein Beispiel

T	**K**	plus	$\Box(A) \rightarrow A$
S4	**T**	plus	$\Box(A) \rightarrow \Box\Box A$
B	**T**	plus	$\neg A \rightarrow \Box\neg\Box A$
S5	**T**	plus	$\neg\Box(A) \rightarrow \Box\neg\Box A$
D	**K**	plus	$\Box(A) \rightarrow \Diamond A$
S4.2	**S4**	plus	$\Diamond\Box(A) \rightarrow \Box\Diamond A$
S4.3	**S4**	plus	$\Box(\Box(A \rightarrow B)) \vee \Box(\Box(B \rightarrow A))$
C	**K**	aber	$\dfrac{A \rightarrow B}{\Box A \rightarrow \Box B}$ anstelle von (G).

Abbildung 3.2: Einige modallogische Systeme

dafür ist in Abbildung 3.3 zusehen.

Es besteht aus Zuständen und Übergängen (Transitionen) zwischen diesen. In den verschiedenen Ausprägungen des Begriffs eines Transitionssystems werden unterschiedliche Einschränkungen an die Menge der möglichen Zustände und ihre Repräsentation gestellt. In der Theorie der erweiterten endlichen Automaten sind sogar unendlich viele Zustände zugelassen. In unserem Beispiel gibt es vier Zustände, deren Bezeichnung keine große Rolle spielt. Außerdem gibt es zwei aussagenlogische Variablen s und r und in jedem Zustand sind die Wahrheitswerte von s und r wie in der Abbildung angegeben festgelegt. In der Regel sind die Übergänge in einem Transitionssystem, also die Kanten in der zugehörigen Diagrammdarstellung, durch zusätzliche Symbole markiert. Für unsere augenblicklichen Zwecke können wir davon abstrahieren. Das angegebene Beispiel beschreibt den Senderautomaten in einem „alternating bit protocol“ und der interessierte Leser kann die Kantenmarkierungen auf Seite 75 in [Holzmann, 1991] nachlesen.
Wir möchten eine Sprache entwickeln, die es erlaubt Eigenschaften von Transitionssystemen zu formulieren mit dem Ziel später auch Methoden zum Nachweis dieser Eigenschaften in einem vorgegebenen System zu entwickeln oder mit dem Ziel eine Logik zu finden, die es erlaubt aus bereits verifizierten Aussagen weitere gültige Aussagen herzuleiten. Dabei möchten wir in dem Beispiel aus Abbildung 3.3 ausdrücken können, daß in jedem Nachfolgezustand eines Zustands, in dem s und r beide wahr sind, jedenfalls s falsch ist. Dasselbe bleibt richtig, wenn wir zwei Transitionen hintereinander ausführen, wird aber falsch bei drei hintereinander ausgeführen Transitionen. Eine duale Aussage gilt, wenn wir in einem Zustand beginnen, in dem s und

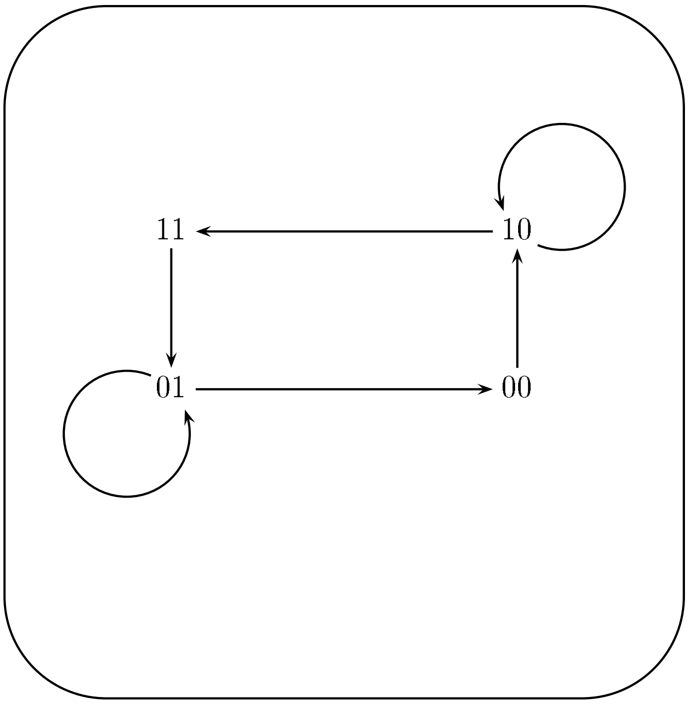


Abbildung 3.3: Beispiel eines Transitionssystems

r beide falsch sind. Aufgrund der Anwesenheit der aussagenlogischen Variablen s und r sollte die gewünschte Repräsentationssprache auf jeden Fall die Aussagenlogik mit umfassen. Zusätzlich führen wir ein Sprachkonstrukt ein, das es erlaubt, die Gültigkeit von Formeln nach Ausführung einer Transition auszudrücken. Genauer soll die Formel $\Box F$ in einem Zustand z_1 wahr sein, wenn in allen Zuständen z, die von z_1 aus in einem Transitionsschritt erreicht werden können, die Formel F wahr ist. Die oben erwähnten Aussagen lassen sich jetzt formal aufschreiben als:

$$\begin{array}{lll}
(s \wedge r) & \rightarrow & \Box\neg s \\
(s \wedge r) & \rightarrow & \Box\Box\neg s \\
(s \wedge r) & \rightarrow & \Box\Box\Box\neg s \\
(\neg s \wedge \neg r) & \rightarrow & \Box s \\
(\neg s \wedge \neg r) & \rightarrow & \Box\Box s \\
(\neg s \wedge \neg r) & \rightarrow & \Box\Box\Box s
\end{array}$$

Diese Überlegungen führen zu den folgenden grundlegenden Definitionen der Modallogik. Die Motivation über Transitionssysteme entspricht nicht der historischen Entwicklung dieser Definitionen, weist dafür aber bereits die Richtung für mögliche Anwendungen. In der historischen Perspektive spielte die Modellierung der Redewendung „Aussage F ist notwendigerweise wahr“ eine wichtige Rolle. Der Vorschlag sie zu ersetzen durch „F ist wahr in allen möglichen Welten“ geht auf Leibniz zurück.

Definition 36 (Kripke Rahmen)
Ein **Kripke Rahmen** (G, R) *besteht aus einer beliebigen nicht leeren Menge* G *und einer zweistelligen Relation* R *auf* G.
Die Elemente von G *heißen* **mögliche Welten**, *die Relation* R *die* **Zugänglichkeitsrelation**.

Definition 37 (Aussagenlogische Kripke Strukturen)
Eine **aussagenlogische Kripke Struktur** $\mathcal{K} = (G, R, v)$ *besteht aus einem Kripke Rahmen* (G, R) *und einer Abbildung* $v : G \times P \rightarrow \{0, 1\}$, *die jeder aussagenlogischen Variablen* $p \in P$ *in Abhängigkeit von einer möglichen Welt* $g \in G$ *einen Wahrheitswert* $v(g, p)$ *zuordnet.*

Kripke Strukturen liefern eine Semantik für die modallogischen Formeln aus Fml_{ALmod}. In Analogie zur Wahrheitsdefinition für die Prädikatenlogik definieren wir die Gültigkeit von modallogischen Formeln in Kripke Strukturen. Die Situation ist hier ein bißchen komplizierter, da wir zwischen der Gültigkeit in einer gegebenen Welt innerhalb einer Struktur und der Gültigkeit in der Struktur als ganzes unterscheiden müssen.

Definition 38 *Sei* $\mathcal{K}= (G, R, v)$ *eine Kripke Struktur,* g *eine Welt in* G *und* $F \in Fml_{ALmod}$*. Die Relation* $(\mathcal{K}, g) \models F$*, gelesen als „* F *ist wahr (oder gültig) in der Welt* g *der Kripke Struktur* $\mathcal{K}$*“, wird induktiv wie folgt definiert. Zur Vereinfachung der Notation schreiben wir* $g \models F$ *anstelle von* $(\mathcal{K}, g) \models F$

1	$g \models p$	*gdw*	$v(g, p) = 1$ *für* $p \in P$
2	$g \models F \wedge H$	*gdw*	$g \models F$ *und* $g \models H$
3	$g \models F \vee H$	*gdw*	$g \models F$ *oder* $g \models H$
4	$g \models F \rightarrow H$	*gdw*	(*nicht* $g \models F$) *oder* $g \models H$
5	$g \models \neg F$	*gdw*	*nicht* $g \models F$
6	$g \models \Box F$	*gdw*	*für alle* h *mit* $R(g, h)$ *gilt* $h \models F$
7	$g \models \Diamond F$	*gdw*	*es gibt ein* h *mit* $R(g, h)$ *so daß* $h \models F$

Die ersten fünf Zeilen dieser Definition wiederholen die Berechnung von Wahrheitswerten für aussagenlogische Formeln, wichtig und neu sind die Zeilen 6 und 7.

Definition 39 *Sei* $\mathcal{K}= (G, R, v)$ *eine Kripke Struktur,* $F \in Fml_{ALmod}$*. Wir sagen,* F *ist wahr (oder gültig) in* $\mathcal{K}$*, in Symbolen* $\mathcal{K} \models F$*, falls* F *in allen Welten von* $\mathcal{K}$*wahr ist, d.h. falls* $(\mathcal{K}, g) \models F$ *gilt für alle* $g \in G$*.*

Beispiel 12 *Ist* $\mathcal{K}_1$ *die durch das Transitionssystem in Abbildung 3.3 eindeutig bestimmte Kripke Struktur, dann gilt:*

$$
\begin{array}{l}
\mathcal{K}_1 \models (s \wedge r) \rightarrow \Box\neg s \\
\mathcal{K}_1 \models (s \wedge r) \rightarrow \Box\Box\neg s \\
\mathcal{K}_1 \not\models (s \wedge r) \rightarrow \Box\Box\Box\neg s \\
\mathcal{K}_1 \models (\neg s \wedge \neg r) \rightarrow \Box s \\
\mathcal{K}_1 \models (\neg s \wedge \neg r) \rightarrow \Box\Box s \\
\mathcal{K}_1 \not\models (\neg s \wedge \neg r) \rightarrow \Box\Box\Box s
\end{array}
$$

Bei der Definition einer modallogischen Konsequenzrelation ergibt sich dieselbe Notwendigkeit wie im Fall der Wahrheitsdefinition zwischen einer lokalen Version, d.h. einer auf eine fixierte Welt bezogenen Version und einer globalen Version zu unterscheiden.

Definition 40 *(Lokale modallogische Konsequenz)*
Sei $\Sigma \subseteq Fml_{ALmod}$ *und* $F \in Fml_{ALmod}$.

$$\Sigma \models_L F$$

gilt genau dann, wenn für jede Kripke Struktur $\mathcal{K} = (G, R, v)$ *und jede Welt* $g \in G$ *gilt:*

$$\textit{falls } g \models \Sigma \text{ , } \textit{dann } g \models F$$

Das Symbol $\models$ wird hier mit anderer Bedeutung als bisher verwendet. Im Kontext von Programmiersprachen würde man sagen, das Symbol $\models$ ist überladen. Die beiden Bedeutungen von $\models$ sind allerdings leicht auseinander zuhalten. In der ersten Bedeutung steht auf der rechten Seite eine Welt einer Kripke Struktur, in der zweiten eine Formelmenge. Wir benutzen diese Notation um das Symbol $\vdash$ zur Bezeichnung der syntaktischen Ableitung frei zu haben.

Typischerweise wird eine modallogische Konsequenzrelation durch einen weiteren Parameter eingeschränkt sein. Die genannte Forderung wird nicht für jede Kripke Struktur verlangt, sondern nur für solche, die in einer bestimmten Klasse $\mathbf{T}$ von Kripke Strukturen liegen. $\mathbf{T}$ könnte z.B. aus allen Kripke Strukturen bestehen, deren Zugänglichkeitsrelation reflexiv ist. Falls eine solche Einschränkung nicht aus dem Zusammenhang heraus deutlich ist, schreiben wir $\models_L^{\mathbf{T}}$ anstelle von $\models_L$.

Definition 41 *(Globale modallogische Konsequenz)*
Sei $\Sigma \subseteq Fml_{ALmod}$ *und* $F \in Fml_{ALmod}$.

$$\Sigma \models_G F$$

gilt genau dann, wenn für jede Kripke Struktur $\mathcal{K} = (G, R, v)$*:*

$$\textit{falls für jede Welt } g \in G \textit{ gilt } g \models \Sigma$$

dann

$$\textit{gilt auch für jede Welt } g \in G \quad g \models F$$

Falls erforderlich schreiben wir wieder $\models_G^{\mathbf{T}}$ *anstelle von* $\models_G$.

Definition 42 *(Tautologien)*
Eine Formel F heißt eine modallogische **Tautologie** *oder* **allgemeingültig**, *wenn* $\emptyset \models F$ *gilt.*
Wir haben den Index für $\models$ *weggelassen, da im Fall einer leeren Prämissenmenge die beiden Konsequenzrelationen zusammenfallen.*
Gilt für eine Klasse $\mathbf{T}$ *von Kripke Strukturen* $\emptyset \models^{\mathbf{T}} F$ *dann nennen wir F eine* $\mathbf{T}$*-Tautologie.*

Eine Formelmenge $\Sigma \subseteq Fml_{ALmod}$ *heißt* **erfüllbar**, *wenn es eine Kripke Stuktur* $\mathcal{K}$*und eine Welt g gibt mit* $(\mathcal{K}, g) \models \Sigma$.

Der Unterschied zwischen der lokalen und der globalen Konsequenzrelation zeigt sich deutlich in der Gültigkeit des Deduktionstheorems.

Lemma 35 *(Deduktionstheorem) Für* $A, B \in Fml_{ALmod}$ *gilt*

$$\begin{array}{lll} B \models_L A & \textit{gdw} & \models_L B \to A \\ & \textit{gdw} & \{B, \neg A\} \textit{ ist nicht erfüllbar} \end{array}$$

Beweis: Durch Einsetzen der Definitionen sieht man sofort

$$B \models_L A \quad \text{gdw} \quad \{B, \neg A\} \text{ ist nicht erfüllbar}$$

Rein aussagenlogisch ist die folgende Äquivalenz gültig:

$$B \to A \text{ ist allgemeingültig} \quad \text{gdw} \quad \{B, \neg A\} \text{ ist nicht erfüllbar}$$

Die folgende Äquivalenz, die wieder durch Einsetzen der Definitionen einzusehen ist, schließt die Argumentationskette:

$$B \to A \text{ ist allgemeingültig} \quad \text{gdw} \quad \models_L B \to A$$

■

3.1.3 Übungsaufgaben

Übungsaufgabe 3.1.1 *Mit* $\mathbf{K}$ *bezeichnen wir die Klasse aller Kripke Strukturen ohne Einschränkung an die Zugänglichkeitsrelation.*

Zeigen Sie, daß $\Box F \leftrightarrow \neg\Diamond\neg F$ *eine* **K**-*Tautologie ist.*
Zeigen, daß auch die folgenden Formeln **K**-*Tautologien sind.*

1. $\Box(P \to Q) \to (\Box P \to \Box Q)$
2. $\Box(P \wedge Q) \leftrightarrow (\Box P \wedge \Box Q)$
3. $\Diamond(P \vee Q) \leftrightarrow (\Diamond P \vee \Diamond Q)$
4. $(\Box P \vee \Box Q) \to \Box(P \vee Q)$
5. $\Diamond(P \wedge Q) \to (\Diamond P \wedge \Diamond Q)$

Übungsaufgabe 3.1.2 *Mit* **S5** *bezeicht man die Klasse aller Kripke Strukturen, deren Zugänglichkeitsrelation eine Äquivalenzrelation ist.*
Geben Sie Kripke Strukturen an, die zeigen, daß die folgenden Formeln nicht einmal **S5**-*Tautologien sind.*

1. $\Box(P \vee Q) \to (\Box P \vee \Box Q)$
2. $(\Diamond P \wedge \Diamond Q) \to \Diamond(P \wedge Q)$

Übungsaufgabe 3.1.3 *Mit* **T** *bezeicht man die Klasse aller Kripke Strukturen, mit reflexiver Zugänglichkeitsrelation, mit* **S4** *die Klasse mit reflexiv-transitivem* R.
Aussagenlogische Formeln der Form $M_1 \ldots M_k \quad p$ *, wobei jedes* M_i *einer der Operatoren* $\Box$ *oder* $\Diamond$ *und* p *eine Aussagenvariabe ist, heißen modale Literale.*
Zeigen Sie:

1. *Die folgenden Beziehungen zwischen modalen Literalen sind* **T**-*Tautologien:*
   (a) $\Box p \to p$
   (b) $p \to \Diamond p$
   (c) $\Box\Box p \to \Box p$
   (d) $\Box\Diamond p \to \Diamond p$
   (e) $\Box p \to \Diamond\Box p$
   (f) $\Diamond p \to \Diamond\Diamond p$

2. *Die folgenden modalen Formeln sind* **S4**-*Tautologien:*

   (a) $\Box\Box p \leftrightarrow \Box p$

   (b) $\Diamond\Diamond p \leftrightarrow \Diamond p$

3. **S5**-*Tautologien:*

   (a) $\Diamond\Box p \leftrightarrow \Box p$

   (b) $\Box\Diamond p \leftrightarrow \Diamond p$

Übungsaufgabe 3.1.4 *Das Deduktionstheorem gilt nicht für die globale Konsequenzrelation. Eine Implikationsrichtung ist aber noch richtig. Welche ist das? Geben Sie ein Gegenbeispiel für die Implikationsrichtung, die nicht mehr wahr ist.*

3.2 Korrespondenztheorie

Ein interessantes und häufig bearbeitetes Teilgebiet der modalen Logik ist die Korrespondenztheorie. Sie versucht Antworten zu finden auf die Frage nach Zusammenhängen zwischen Eigenschaften eines Rahmens (G, R) und Formeln der modalen Aussagenlogik, die in allen Kripke Strukturen mit diesem Rahmen gelten. Wir beginnen mit einem einfachen Beispiel.

Lemma 36
Eine Rahmen (G, R) ist reflexiv genau dann, wenn in allen Kripke Strukturen $\mathcal{K} = (G, R, v)$ die Formel $\Box p \rightarrow p$ gültig ist.

Dabei heißt ein Rahmen (G, R) reflexiv, wenn die zugehörige binäre Relation R reflexiv ist, d.h. $(G, R) \models \forall x R(x, x)$ gilt. Der Buchstabe p bezeichnet hier wie in allen nachfolgenden Formeln eine ausagenlogische Variable.
Beweis:
Wir nehmen zunächst an, daß (G, R) reflexiv ist und betrachten eine beliebige Funktion $v : G \times P \rightarrow \{0, 1\}$. Für jedes $g \in G$ müssen wir $(\mathcal{K}, g) \models \Box p \rightarrow p$ zeigen. Das ist trivialerweise richtig, falls $(\mathcal{K}, g) \not\models \Box p$. Gilt aber $(\mathcal{K}, g) \models \Box p$, so folgt nach Definition für alle $h \in G$ mit $R(g, h)$ auch $(\mathcal{K}, h) \models p$. Da R reflexiv ist gilt insbesondere $R(g, g)$ und damit, wie gewünscht, $(\mathcal{K}, g) \models p$.
Zum Beweis der umgekehrten Richtung nehmen wir an, daß (G, R) nicht reflexiv ist, und wir wählen ein $g_0 \in G$ mit $\neg R(g_0, g_0)$. Wir definieren eine Funktion v_0 durch

$$v_0(g, p) = \begin{cases} 1 & \text{falls } R(g_0, g) \\ 0 & \text{sonst.} \end{cases}$$

Offensichtlich gilt für $\mathcal{K}_0 = (G, R, v_0)$ zunächst $(\mathcal{K}_0, g_0) \models \Box p$, aber auch $(\mathcal{K}_0, g_0) \models \neg p$. Also ist $\Box p \rightarrow p$ in $\mathcal{K}_0$ nicht wahr.

■

Definition 43 *Sei $\mathcal{G}$ eine Klasse von Rahmen und $F \in Fml_{ALmod}$.*
Wir sagen F **charakterisiert** *die Klasse $\mathcal{G}$, falls für jeden Rahmen (G, R) gilt:*

$(G, R) \in \mathcal{G}$
gdw
für jedes v gilt $(G, R, v) \models F$

In dieser Terminologie besagt Lemma 36, daß die Formel $\Box p \rightarrow p$ die Klasse aller reflexiven Rahmen charakterisiert. Hier ist ein weiteres Charakterisierungsresultat.

Lemma 37
Die Formel $\Box p \rightarrow \Box\Box p$ charakterisiert die Klasse aller transitiven Rahmen.

Beweis:
Wir nehmen zunächst an, daß (G, R) transitiv ist und daß für ein beliebiges v und $g \in G$ gilt für $\mathcal{K} = (G, R, v)$ $(\mathcal{K}, g) \models \Box p$. Wir müssen $(\mathcal{K}, g) \models \Box\Box p$ zeigen. Das bedeutet, wir müssen für alle Welten g_1, g_2 mit $R(g, g_1)$ und $R(g_1, g_2)$ zeigen, daß $(\mathcal{K}, g_2) \models p$ gilt. Wegen der Transitivität von R gilt auch $R(g, g_2)$, damit liefert die Voraussetzung $(\mathcal{K}, g) \models \Box p$ unmittelbar $(\mathcal{K}, g_2) \models p$.
Zum Beweis der umgekehrten Richtung nehmen wir wieder an, daß (G, R) nicht transitiv ist, d.h. es gibt $g_0, g_1, g_2 \in G$ mit $R(g_0, g_1)$, $R(g_1, g_2)$ und $\neg R(g_0, g_2)$. Wir definieren eine Funktion v_0 wie in Lemma 36 durch

$$v_0(g, p) = \begin{cases} 1 & \text{falls } R(g_0, g) \\ 0 & \text{sonst.} \end{cases}$$

Aus dieser Definition folgt für $\mathcal{K}_0 = (G, R, v_0)$ unmittelbar $(\mathcal{K}_0, g_0) \models \Box p$. Da andererseits $(\mathcal{K}_0, g_2) \models \neg p$ gilt, folgt $(\mathcal{K}_0, g_0) \not\models \Box\Box p$. Zusammengenommen haben wir gezeigt, daß die Formel $\Box p \rightarrow \Box\Box p$ in $\mathcal{K}_0$ nicht wahr ist.

■

Nach diesen Einzelbeispielen mag man sich fragen, ob es nicht allgemeinere Resultate gibt. Aber wie sollten diese aussehen? Was kann man überhaupt erwarten? Gefragt wird nach Korrespondenzen zwischen Formeln der modalen Aussagenlogik auf der einen Seite und Klassen von Rahmen auf der anderen Seite. Die modallogischen Formeln sind in der Definition von Fml_{ALmod}eindeutig festgelegt, aber welche Klassen von Rahmen wollen wir betrachten? Sicherlich können wir nicht alle zulassen. Selbst wenn wir uns auf Mengen abzählbarer Rahmen beschränken, gibt es davon überabzählbar viele, zuviele, um eine Korrespondenz zur abzählbaren Menge Fml_{ALmod}herstellen zu können. In allen bisherigen Einzelresultaten kamen nur Klassen von Rahmen vor, die durch eine Formel des Prädikatenkalküls erster Stufe beschrieben wurden. Genügt das? oder sind das vielleicht schon zuviele? Genauer formuliert lauten diese Fragen:

1	reflexiv	$\forall x R(x, x)$
2	symmetrisch	$\forall x \forall y (R(x, y) \rightarrow R(y, x))$
3	seriell	$\forall x \exists y R(x, y)$
4	transitiv	$\forall x \forall y \forall z (R(x, y) \wedge R(y, z) \rightarrow R(x, z))$
5	Euklidisch	$\forall x \forall y \forall z (R(x, y) \wedge R(x, z) \rightarrow R(y, z))$
6	schwach funktional	$\forall x \forall y \forall z (R(x, y) \wedge R(x, z) \rightarrow y = z)$
7	funktional	schwach funktional und $\forall x \exists y R(x, y)$
8	dicht	$\forall x \forall y (R(x, y) \rightarrow \exists z (R(x, z \wedge R(z, y))$
9	schwach zusammenhängend	$\forall x \forall y \forall z (R(x, y) \wedge R(x, z) \rightarrow (R(y, z) \vee y = z \vee R(z, y))$
10	schwach gerichtet	$\forall x \forall y \forall z ((R(x, y) \wedge R(x, z)) \rightarrow \exists w (R(y, w) \wedge R(z, w))$
11	konfluent	$\forall x \forall y (R(x, y) \rightarrow \exists z (R(x, z) \wedge R(y, z)))$

Abbildung 3.4: Einige Eigenschaften für Rahmen

1	reflexiv	$\Box p \rightarrow p$
2	symmetrisch	$p \rightarrow \Box \Diamond p$
3	seriell	$\Box p \rightarrow \Diamond p$
4	transitiv	$\Box p \rightarrow \Box \Box p$
5	Euklidisch	$\Diamond p \rightarrow \Box \Diamond p$
6	schwach funktional	$\Diamond p \rightarrow \Box p$
7	funktional	$\Diamond p \leftrightarrow \Box p$
8	dicht	$\Box \Box p \rightarrow \Box p$
9	schwach zusammenhängend	$\Box((p \wedge \Box p) \rightarrow q) \vee \Box((q \wedge \Box q) \rightarrow p)$
10	schwach gerichtet	$\Diamond \Box p \rightarrow \Box \Diamond p$
11	konfluent	$\Diamond \Box p \rightarrow \Diamond p$

Abbildung 3.5: Einige Charakterisierungsresultate

- Gibt es zu jeder Formel ϕ des Prädikatenkalküls erster Stufe eine Formel $F \in Fml_{ALmod}$, so daß F die durch ϕ definierte Klasse von Rahmen charakterisiert?
- Gibt es zu jeder Formel $F \in Fml_{ALmod}$ eine Formel ϕ des Prädikatenkalküls erster Stufe, so daß F die durch ϕ definierte Klasse von Rahmen charakterisiert?

Wir stellen die Beantwortung beider Fragen noch etwas zurück und betrachten zunächst ein allgemeines Korrespondenzresultat. Es betrifft modallogische Formeln der Form $\Diamond^m \Box^n p \rightarrow \Box^j \Diamond^k p$ für natürliche Zahlen m, n, j, k, die auch $= 0$ sein können. Dabei steht

$$\Box^n p \text{ für } \underbrace{\Box \ldots \Box}_{n- \text{ mal}} p$$

Insbesondere $\Box^0 p$ für p. Um die korrespondierende prädikatenlogische Formel übersichtlich aufschreiben zu können brauchen wir die Abkürzung:

$$R^n(x, y) = \exists u_1 \ldots \exists u_{n-1}(R(x, u_1) \wedge \ldots \wedge R(u_i, u_{i+1}) \wedge \ldots \wedge R(u_{n-1}, y)).$$

Für $n = 1$ soll $R^1(x, y) = R(x, y)$ bedeuten und für $n = 0$ setzen wir $R^0(x, y) = x \doteq y$.

Definition 44

$$\begin{aligned} C(m, n, j, k) \quad = \quad & \forall w_1 \forall w_2 \forall w_3((R^m(w_1, w_2) \wedge R^j(w_1, w_3)) \\ & \rightarrow \exists w_4(R^n(w_2, w_4) \wedge R^k(w_3, w_4))) \end{aligned}$$

Die Formel $C(m, n, j, k)$ beschreibt eine Art Konfluenzeigenschaft, siehe Abbildung 3.6.

Satz 38

Für jedes Quintuple natürlicher Zahlen m, n, j, k charakterisiert die Formel $\Diamond^m \Box^n p \rightarrow \Box^j \Diamond^k p$, die Klasse aller Rahmen, in denen $C(m, n, j, k)$ wahr ist.

Beweis: Der folgende Beweis benutzt stillschweigend die in Übungsaufgabe 3.2.3 genannten Beziehungen.
Sei zuächst eine Rahmen (G, R) gegeben mit $(G, R) \models C(m, n, j, k)$. Für jede Kripke Struktur $\mathcal{K} = (G, R, v)$ und jedes $g \in G$, für das $(\mathcal{K}, g) \models \Diamond^m \Box^n p$

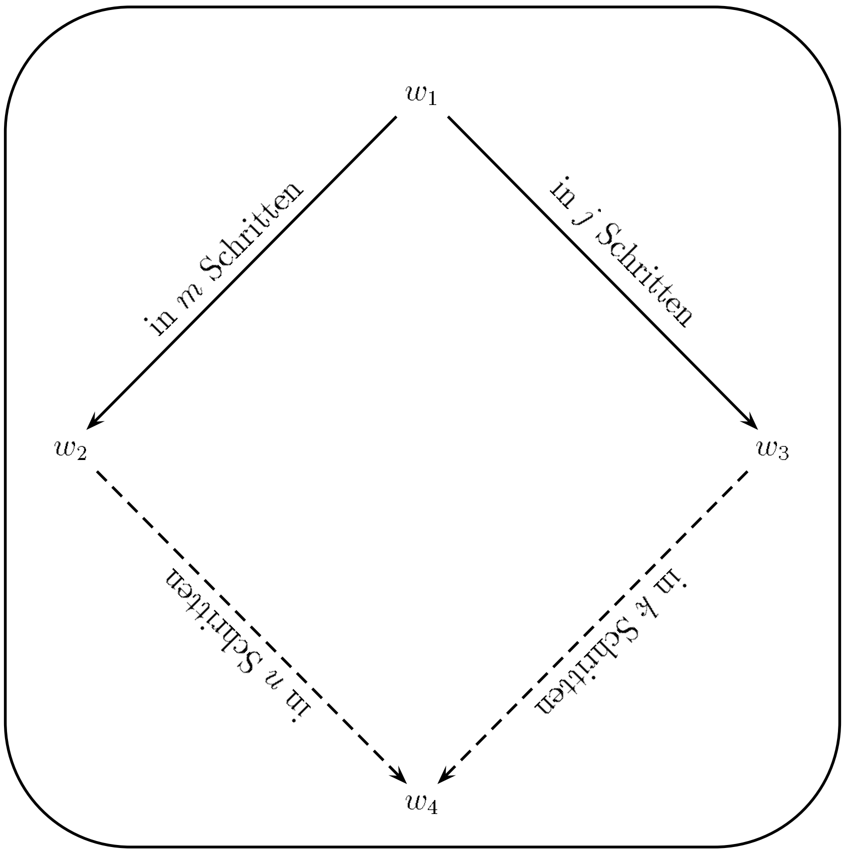


Abbildung 3.6: Visualisierung der Formel $C(m, n, j, k)$

gilt müssen wir $(\mathcal{K}, g) \models \Box^j \Diamond^k p$ zeigen. Aus $(\mathcal{K}, g) \models \Diamond^m \Box^n p$ folgt die Existenz einer Welt $g_1 \in G$ mit $(G, R) \models R^m(g, g_1)$ und $(\mathcal{K}, g_1) \models \Box^n p$. Um $(\mathcal{K}, g) \models \Box^j \Diamond^k p$ nachzuweisen müssen wir für eine beliebige Welt $g_2 \in G$ mit $(G, R) \models R^j(g, g_2)$ zeigen, daß $(\mathcal{K}, g_2) \models \Diamond^k p$ gilt. Wegen der Annahme $(G, R) \models C(m, n, j, k)$ gibt es auf jeden Fall einmal eine Welt $h \in G$ mit $(G, R) \models R^n(g_1, h)$ und $(G, R) \models R^k(g_2, h)$. Wegen $(\mathcal{K}, g_1) \models \Box^n p$ gilt $v(h, p) = 1$. Zusammen mit $R^k(g_2, h)$ folgt daraus $(\mathcal{K}, g_2) \models \Diamond^k p$.
Sei jetzt umgekehrt ein Rahmen (G, R) vorgelegt, so daß für jedes v gilt $(G, R, v) \models \Diamond^m \Box^n p \to \Box^j \Diamond^k p$. Um $C(m, n, j, k)$ zu zeigen, fixieren wir Welten $g, g_1, g_2 \in G$ mit $(G, R) \models R^m(g, g_1)$ und $(G, R) \models R^j(g, g_2)$. Gesucht ist ein *konfluentes* $h \in G$. Dazu definieren wir ein v durch:

$$v(g, p) = \begin{cases} 1 & \text{falls } R^n(g_1, g) \\ 0 & \text{sonst.} \end{cases}$$

Für $\mathcal{K} = (G, R, v)$ gilt nach dieser Definition $(\mathcal{K}, g_1) \models \Box^n p$ also auch $(\mathcal{K}, g) \models \Diamond^m \Box^n p$. Nach Voraussetzung gilt dann auch $(\mathcal{K}, g) \models \Box^j \Diamond^k p$ und damit auch $(\mathcal{K}, g_2) \models \Diamond^k p$. Es existiert also also ein $h \in G$ mit $R^k(g_2, h)$ und $v(g_2, p) = 1$. Nach Definition von v muß dann aber auch $R^n(g_1, h)$ gelten.

■

In [Sterling, 1992] und [Hughes & Cresswell, 1984] findet man ähnliche Beweise. Einen Charakterisierungssatz für eine umfassendere Klasse modallogischer Formeln findet man in [Sahlqvist, 1975]. Eine kurze Darstellung der Sahlqvistschen Theorie findet man auch in Appendix B von [Marx & Venema, 1997].

Wir kehren zur Frage zurück, ob jede durch eine Formel des Prädikatenkalküls erster Stufe definierbare Klasse von Rahmen auch durch eine Formel der modalen Aussagenlogik charakterisiert werden kann. Die Antwort ist negativ. Wir führen in einem Fall einen kompletten Beweis vor und erwähnen danach noch einige ähnlich gelagerte Beispiele. Ein Beweis, daß etwas **nicht** möglich ist, ist in der Regel aufwendiger zu führen, als der Nachweis, daß etwas möglich ist. So brauchen wir auch für das vorliegende Problem erst einen Anlauf: wir geben ein Konstruktionsverfahren an, das jeder Kripke Struktur $\mathcal{K}$ eine Kripke Struktur $\mathcal{K}^*$ zuordnet, so daß in $\mathcal{K}$ und $\mathcal{K}^*$ dieselben Formeln aus Fml_{ALmod} gelten. Wenn wir dann eine spezielle Struktur $\mathcal{K}_1 = (G_1, R_1, v_1)$ finden können, so daß $(G_1, R_1, v_1) \not\models \forall x \neg R(x, x)$ aber $(G_1^*, R_1^*, v_1^*) \models \forall x \neg R(x, x)$, dann kann die Klasse aller irreflexiver Rahmen nicht durch eine Formel aus Fml_{ALmod} charakterisiert werden.

Definition 45 *Gegeben sei eine Kripke Struktur $\mathcal{K} = (G, R, v)$. Für jedes $g \in G$ seien g^1, g^2 zwei neue, verschiedene Welten. Die Kripke Struktur $\mathcal{K}^* = (G^*, R^*, v^*)$ wird definiert durch*

$$\begin{array}{lll l} G^* & = & \{g^i \mid g \in G, i \in \{1,2\}\} & \\ R^*(g^i, h^j) & gdw & R(g,h) \& i,j \in \{1,2\} & \textit{falls } g \neq h \\ R^*(g^i, g^j) & gdw & R(g,g) \& i \neq j & \\ v^*(g^i, p) & = & v(g,p) & \end{array}$$

Wichtig bei dieser Definition ist, daß in keinem Fall $R^*(g^1, g^1)$ oder $R^*(g^2, g^2)$ gelten kann. Falls $\neg R(g,g)$, dann gilt überhaupt keine der vier möglichen R^*-Relationen zwischen den Elementen $\{g_1, g_2\}$. Falls $R(g,g)$, dann gelten genau $R^*(g^1, g^2)$ und $R^*(g^2, g^1)$.

Lemma 39
Für jede Kripke Struktur $\mathcal{K}$ und jede Welt g von $\mathcal{K}$ gilt für jede Formel $F \in Fml_{ALmod}$:

$$\begin{array}{llll} 1 & (\mathcal{K}, g) \models F & gdw & (\mathcal{K}^*, g^1) \models F \\ 2 & (\mathcal{K}, g) \models F & gdw & (\mathcal{K}^*, g^2) \models F \\ 3 & \mathcal{K} \models F & gdw & \mathcal{K}^* \models F \end{array}$$

Beweis
Wir beweisen 1 und 2 simultan durch Induktion über den Formelaufbau von F. Behauptung 3 ist dann eine unmittelbare Konsequenz aus 1 und 2.
Im Induktionsanfang ist $F = p$ für eine aussagenlogische Variable p und die beiden Behauptungen folgen unmittelbar aus der Definition von v^*. Die Induktionsschritte für die aussagenlogischen Konnektoren sind einfach. Wir betrachten hier nur den Fall $F = \Box F_1$, der Fall $F = \Diamond F_1$ kann analog abgehandelt werden. Gelte also $(\mathcal{K}, g) \models \Box F_1$ und wir wollen $(\mathcal{K}^*, g^i) \models \Box F_1$ zeigen. Aus der Annahme folgt

$$(\mathcal{K}, h) \models F_1 \text{ für alle } h \in G \text{ mit } R(g,h)$$

Um $(\mathcal{K}^*, g^i) \models \Box F_1$ zu zeigen, müssen wir alle $u \in G^*$ betrachten mit $R^*(g^i, u)$. Hier unterscheidet man am besten zwei Fälle.

Erster Fall: $u = h^j$ für ein $h \in G$, ein $j \in \{1, 2\}$ mit $g \neq h$ und $R(g, h)$. Nach Annahme gilt $(\mathcal{K}, h) \models F_1$ und die Induktionsvoraussetzung liefert daraus $(\mathcal{K}^*, h^j) \models F_1$.
Zweiter Fall: $u = g^j$ und $R(g, g)$. Nach Annahme gilt dann $(\mathcal{K}, g) \models F_1$ und die Induktionsvoraussetzung liefert wieder $(\mathcal{K}^*, g^j) \models F_1$.
Man beachte, daß im zweiten Fall $i \neq j$ sein muss und es an dieser Stelle wichtig ist, die Teilaussagen 1 und 2 des Lemmas simultan zu beweisen.

Zum Beweis der umgekehrten Implikationen elte jetzt etwa $(\mathcal{K}^*, g^1) \models \Box F_1$ und wir wollen $(\mathcal{K}, g) \models \Box F_1$ zeigen. (Der Fall $(\mathcal{K}^*, g^2) \models \Box F_1$ wird natürlich völlig analog behandelt.) Für jedes $h \in G$ mit $R(g, h)$ müssen wir also $(\mathcal{K}, h) \models F_1$ zeigen. Nach Definition von R^* gilt dann $R^*(g^1, h^2)$ unabhängig davon ob $g = h$ oder $g \neq h$ ist, nach Annahme gilt also auch $(\mathcal{K}^*, h^2) \models F_1$, woraus mit Hilfe der Induktionsvoraussetzung $(\mathcal{K}, h) \models F_1$ folgt.

■

Satz 40 *Die Klasse aller irreflexiven Rahmen, d.h. die Klasse aller Rahmen (G, R), in denen die Formel $\forall x \neg R(x, x)$ gilt, kann nicht durch eine Formel der modalen Aussagenlogik charakterisiert werden.*

Beweis
Angenommen $F \in Fml_{ALmod}$ wäre eine charakterisierende Formel. Wir betrachten den Rahmen (G_1, R_1) der nur eine Welt g_0 enthält und in dem $R_1(g_0, g_0)$ gilt. Nach Konstruktion (in Definition 45) ist (G^*, R^*) irreflexiv. Für ein beliebiges v sollte also $(G^*, R^*, v^*) \models F$ gelten, aber $(G, R, v) \models F$ nicht. Lemma 39 widerspricht dem: ein F mit dieser Eigenschaften kann es nicht geben.

■

Weitere Beispiele nicht charakterisierbarer Klassen von Rahmen sind in Abbildung 3.7 angegeben. Weitergehende Resultate findet man in [van Benthem, 1984]. Schließlich kommen wir noch auf die Frage zurück, ob es zu jeder Formel $F \in Fml_{ALmod}$ eine Formel ϕ des Prädikatenkalküls erster Stufe gibt, so daß F die durch ϕ definierte Klasse von Rahmen charakterisiert. Auch hier ist die Antwort negativ. Im allgemeinen ist die Klasse aller Rahmen, die durch F charakterisiert wird, nur durch eine Formel in der Prädikatenlogik zweiter Stufe definierbar. Die bisher betrachteten Beispiele sind also irreführende Spezialfälle.

$\forall x \neg R(x,x)$	Irreflexivität	
$\forall x \forall y R(x,y)$	Universalität	siehe [Fitting, 1993]
$\exists x \exists y R(x,y)$	Nichttrivialität	siehe [Fitting, 1993]
$\forall x \exists y (R(x,y) \wedge R(y,y))$		siehe [van Benthem, 1984]

Abbildung 3.7: Beispiele nicht charakterisierbarer Klassen

Lemma 41
Innerhalb der Klasse aller transitiven Rahmen charakterisiert die Formel

$$\Box(\Box p \rightarrow p) \rightarrow \Box p$$

die Klasse aller transitiven Rahmen (G, R)*, so daß keine unendliche aufsteigende Kette* $g_0, g_1, \ldots, g_n \ldots$ *in* G *existiert mit:*

$$R(g_0, g_1), R(g_1, g_2), \ldots, R(g_n, g_{n+1}), \ldots$$

Mit anderen Worten: die Relation R^{-1} *ist wohlfundiert.*

Beweis: siehe [van Benthem, 1984], Seite 195ff und Aufgabe 3.2.12.

■

Ein weiteres interessantes Beispiel läßt sich in der Formelklasse einer multimodalen Logik finden. In einer multimodale Logik kommt nicht nur ein Box-operator $\Box$ vor, sondern es sind mehrere zugelassen, in unserem Beispiel sind das $\Box_a$ und $\Box_b$. Ein Rahmen für eine multimodale Logik enthält nicht nur eine Zugänglichkeitsrelation R sondern entsprechend mehrere, in unserem Beispiel sind das R_a und R_b.

Definition 46
Die **transitive Hülle** *einer zweistelligen Relatione* R *ist die kleinste Relation* R_t *mit den Eigenschaften:*

1. R_t *ist transitiv und symmetrisch*
2. $R \subseteq R_t$

Satz 42
Die Konjunktion der beiden multimodalen Formeln

$$\Box_a p \to (p \wedge \Box_a \Box_b p)$$
$$(\Box_a(p \to \Box_b p) \to (p \to \Box_a p))$$

charakterisiert die Klasse aller multimodalen Rahmen (G, R_a, R_b), so daß R_a die transitive Hülle von R_b ist.

Beweis:
Ist $\mathcal{K} = (G, R_a, R_b, v)$ eine multimodale Kripke Struktur, so daß R_a die transitive Hülle von R_b ist, dann läßt sich leicht nachrechnen, daß die beiden Formeln in $\mathcal{K}$ wahr sind.
Betrachten wir jetzt einen Rahmen (G, R_a, R_b), in dem beide Formeln gelten. Wir wollen zeigen, daß dann R_a die transitive Hülle von R_b ist.
Teil 1:
Wir betrachten $g, h \in G$ mit $R_a(g, h)$ und zeigen, daß dann das Paar (g, h) in der transitiven Hülle $tr(R_b)$ von R_b liegen muß. Dazu definieren wir eine Belegung v wie folgt:

$$v(w, p) = \begin{cases} 1 & \text{falls } (g, w) \text{ in } tr(R_b) \text{ liegt.} \\ 0 & \text{sonst.} \end{cases}$$

Wir wollen uns zuerst davon überzeugen, daß für $\mathcal{K} = (G, R_a, R_b, v)$ gilt:

$$(\mathcal{K}, g) \models \Box_a(p \to \Box_b p)$$

Wir betrachten dazu ein $w \in G$ mit $R_a(g, w)$ und $(\mathcal{K}, w) \models p$. Das ist nach Definition von v nur möglich, wenn (g, w) in der transitiven Hülle von R_b liegt. Ist w' eine Welt mit $R_b(w, w')$ dann liegt natürlich auch (g, w') wieder in $tr(R_b)$ und nach Definition von v folgt $(\mathcal{K}, w') \models p$. Das zeigt $(\mathcal{K}, w) \models \Box_b p$. Da wir angenommen haben, daß $(\Box_a(p \to \Box_b p) \to (p \to \Box_a p))$ in $\mathcal{K}$ gilt, folgt jetzt auch $(\mathcal{K}, g) \models p \to \Box_a p$. Da jede transitive Hülle insbesondere reflexiv ist gilt $(\mathcal{K}, g) \models p$, woraus also auch $(\mathcal{K}, g) \models \Box_a p$ folgt. Nach Annahme über die Welt h ergibt sich daraus $(\mathcal{K}, h) \models p$, was nach Definition von v nur möglich ist, wenn (g, h) in $tr(R_b)$ liegt. Das beendet Teil 1.
Teil 2:
Hier betrachten wir ein Paar von Welten (g, h) in der transitiven Hülle von R_b und wir wollen zeigen, daß $R_a(g, h)$ gilt. Dazu brauchen wir die folgende Belegung:

$$v(w, p) = \begin{cases} 1 & \text{falls } R_a(g, w) \\ 0 & \text{sonst.} \end{cases}$$

Nach Voraussetzung gilt für $\mathcal{K} = (G, R_a, R_b, v)$ jedenfalls $(\mathcal{K}, g) \models \Box_a p \to (p \wedge \Box_a \Box_b p)$. Nach Wahl von v ist die Prämisse der Implikation erfüllt, weswegen auch

$$(*) \qquad (\mathcal{K}, g) \models p \wedge \Box_a \Box_b p$$

gilt. Nach Annahme soll (g, h) in $tr(R_b)$ liegen, es gibt also Welten $g_0, g_1, \ldots, g_k$ mit $R_b(g_i, g_{i+1})$ für alle $0 \leq i < k$ und $g_0 = g$ und $g_k = h$. Wir zeigen

$$(\mathcal{K}, g_i) \models p$$

durch Induktion nach i. Für $i = 0$ ist die Behauptung schon in $(*)$ enthalten. Gelte $(\mathcal{K}, g_i) \models p$. Nach Definition von v folgt daraus $R_a(g, g_i)$ und somit gilt nach $(*)$ auch $(\mathcal{K}, g_{i+1}) \models p$. Aus $(\mathcal{K}, g_k) \models p$, d.h. aus $(\mathcal{K}, h) \models p$, folgt schließlich nach Definition von v wie gewünscht $R_a(g, h)$.

■

Korollar 43
Die durch die Konjunktion der beiden multimodalen Formeln

$$\Box_a p \to (p \wedge \Box_a \Box_b p)$$
$$(\Box_a(p \to \Box_b p) \to (p \to \Box_a p))$$

charakterisierte Klasse multimodalen Rahmen (G, R_a, R_b) ist nicht durch eine Formel der Prädikatenlogik erster Stufe definierbar.

Beweis:
Durch eine Standardanwendung des Kompaktheitssatzes läßt sich zeigen, daß die transitive Hülle einer Relation nicht in der Prädikatenlogik erster Stufe ausgedrückt werden kann.

■

3.2.1 Eine Konstruktionsmethode

Für die Richtung des Charakterisierungsproblems von modalen Formeln zu einer Formel der Prädikatenlogik erster Stufe läßt sich ein Konstruktionsverfahren angeben, in das sich alle bekannten Ergebnisse zusammenfassend einordnen lassen. Wir beschreiben in diesem Abschnitt die Details dieses Ansatzes, der schon in [Ackermann, 1935] publiziert wurde. Anstelle der rein mathematischen Formulierung in [Ackermann, 1935] benutzen wir eine Darstellung, die sich anlehnt an Begriffe aus dem Resolutionskalkül. Siehe auch [Szałas, 1993].

Wir gehen aus von einer modalen Formel A und suchen eine prädikatenlogische Formel F, so daß für jeden Kripke Rahmen (G, R) gilt:

$$\text{für alle } v \text{ gilt } (G, R, v) \models A \text{ gdw } (G, R) \models F$$

Wie wir inzwischen wissen, muß ein F mit dieser Eigenschaft nicht immer existieren. Die Methode, die wir jetzt schildern wollen, wird also auch die Möglichkeiten des Fehlschlags in sich bergen.

Wir brauchen einige Vorbereitungen.

Definition 47

Sei $\mathcal{M}_1$ eine prädikatenlogische Struktur zum Vokabular V_1, $\mathcal{M}_2$ eine Struktur zum Vokabular V_2.

$\mathcal{M}_2$ heißt eine Fortsetzung *von $\mathcal{M}_1$, wenn die Interpretation von Symbolen aus $V_1 \cap V_2$ in $\mathcal{M}_1$ und $\mathcal{M}_2$ übereinstimmen.*

$\mathcal{M}_2$ kann sich also von $\mathcal{M}_1$ nur unterscheiden in der Interpretation von Symbolen die in V_2 aber nicht in $V_1 \cap V_2$ liegen. Insbesondere haben $\mathcal{M}_1$ und $\mathcal{M}_2$ dasselbe Universum.

Definition 48

Seien F_1, F_2 Formeln der Logik zweiter Stufe wobei F_1 im Vokabular V_1 aufgeschrieben ist und F_2 im Vokabular V_2. F_1, F_2 heißen

1. erfüllbarkeitsäquivalent *(e-äquivalent), wenn eine prädikatenlogische V_1-Struktur $\mathcal{M}_1$ und Variablenbelegungen β_1, γ_1 mit $(\mathcal{M}_1, \beta_1, \gamma_1) \models F_1$ genau dann existieren, wenn auch eine V_2-Struktur $\mathcal{M}_2$ und β_2, γ_2 existieren mit $(\mathcal{M}_2, \beta_2, \gamma_2) \models F_2$,*

2. strikt erfüllbarkeitsäquivalent *(se-äquivalent), wenn für jede V_1-Struktur $\mathcal{M}$ und Variablenbelegung β, γ mit $(\mathcal{M}, \beta, \gamma) \models F_1$ eine V_2-Fortsetzung $\mathcal{M}^*$ von $\mathcal{M}$ existiert mit $(\mathcal{M}^*, \beta, \gamma) \models F_2$ und umgekehrt zu jeder V_2-Struktur $\mathcal{N}$ und Variablenbelegung β,γ mit $(\mathcal{N}, \beta, \gamma) \models F_2$ eine V_1-Fortsetzung $\mathcal{N}^*$ von $\mathcal{N}$ existiert mit $(\mathcal{N}^*, \beta, \gamma) \models F_1$.*

Das nächste Lemma sammelt die Eigenschaften der se-Äquivalenz, die wir im folgenden brauchen werden.

Lemma 44

1. *Sind F_1, F_2 Formeln der Logik zweiter Stufe im selben Vokabular $V = V_1 = V_2$, dann sind F_1 und F_2 genau dann se-äquivalent, wenn $F_1 \leftrightarrow F_2$ allgemeingültig ist.*

2. *Sei F eine Formel im Vokabular $V = V_0 \cup \{c\}$, c ein Konstantensymbol. $F_0(x)$ entstehe aus F indem jedes Vorkommen von c in F durch das neue Variablensymbol x ersetzt wird. Dann sind F und $\exists x F_0(x)$ se-äquivalent.*

3. *Sei F eine Formel im Vokabular $V = V_0 \cup \{p\}$, p ein einstelliges Relationsymbol. $F_1(X)$ entstehe aus F indem jedes Vorkommen von $p(t)$ in F durch $X(t)$ mit einem neuen Variablensymbol X ersetzt wird. Dann sind F und $\exists X F_1(X)$ se-äquivalent.*

4. *F_{sk} entstehe aus F durch Skolemisierung. Dann sind F_{sk} und F se-äquivalent.*

5. *Sei K eine Menge von Klauseln, K_1, K_2 zwei Klauseln, p ein Prädikatensymbol und $t_1^1, \ldots t_n^1$ Variable oder Konstantensymbole und $t_1^2, \ldots t_n^2$ beliebige Terme. Es sei σ ein Unifikator für $p(t_1^1, \ldots t_n^1)$ und $p(t_1^2, \ldots t_n^2)$. Außerdem komme p weder in K, K_1 noch K_2 vor. Dann sind*

$$K \wedge (K_1 \vee p(t_1^1, \ldots t_n^1)) \wedge (K_2 \vee \neg p(t_1^2, \ldots t_n^2))$$

   und

$$K \wedge \sigma(K_1 \vee K_2)$$

   se-äquivalent. Für eine korrekte Benutzung der Definition 48 sind die universellen Quantoren einer Klausel, die häufig weggelassen werden, explizit hinzuschreiben.

Beweis zu 1: Sei $\mathcal{M}$ eine beliebige V-Struktur und β, γ beliebige Belegungsfunktionen für die Individuen- bzw. Mengenvariablen und gelte $(\mathcal{M}, \beta, \gamma) \models F_1$. Nach Definition der se-Äquivalenz gibt es eine Erweiterung $\mathcal{M}^*$ von $\mathcal{M}$ mit $(\mathcal{M}^*, \beta, \gamma) \models F_2$. Wegen der Übereinstimmung der Vokabulare muß aber $\mathcal{M} = \mathcal{M}^*$ gelten. Also gilt $(\mathcal{M}, \beta, \gamma) \models F_2$

Beweis zu 2: Aus $(\mathcal{M}, \beta, \gamma) \models F$ folgt unmittelbar $(\mathcal{M}, \beta, \gamma) \models \exists x F_0(x)$. Gilt umgekehrt für eine V_0-Struktur $\mathcal{M}_0$ $(\mathcal{M}_0, \beta, \gamma) \models \exists x F_0(x)$, dann gibt es ein Element $m \in M$ mit $(\mathcal{M}_0, \beta_x^m, \gamma) \models F_0(x)$. Definiert man die Fortsetzung $\mathcal{M}$ von $\mathcal{M}_0$ auf das Vokabular V durch $I(\mathcal{M})(c) = m$, dann gilt offensichtlich $(\mathcal{M}, \beta, \gamma) \models F$.

Beweis zu 3: Wie zu 2

Beweis zu 4: Bekannt.

Beweis zu 5: Wir schreiben die Voraussetzungen etwas detailierter auf. $A_0 = \forall \bar{x} \bigwedge K$, $A_1 = \forall \bar{y}(K_1 \vee p(t_1^1, \dots t_n^1))$, $A_2 = \forall \bar{z}(K_2 \vee \neg p(t_1^2, \dots t_n^2))$ $B = \forall \bar{w} \sigma(K_1 \vee K_2)$ Wir können annehmen, daß die Variablentupel $\bar{x}$,$\bar{y}$, $\bar{z}$ disjunkt sind und keine freien Variablen vorkommen. Der allgemeinste Unifikator σ ist in der vorliegenden Situation besonders einfach:

$$\sigma(v) = \begin{cases} \mathcal{M}(t_j^2) \text{ falls } v = t_j^1 \text{ eine Variable} \\ \mathcal{M}(t_j^1) \text{ falls } v \text{ in } \bar{z}. \end{cases}$$

Die vorausgesetzte Existenz eines Unifikators garantiert, daß diese Definition nicht widersprüchlich ist.

Es ist zu zeigen, daß $A_0 \wedge A_1 \wedge A_2$ und $A_0 \wedge B$ se-äquivalent sind. Dabei besitzt die letzte Formel das Vokabular V und die erste das Vokabular $V \cup \{p\}$. Nur eine Richtung dieses Beweises ist nicht-trivial: Sei $\mathcal{M}$ eine V-Struktur mit $\mathcal{M} \models A_0 \wedge B$. Wir müssen eine Fortsetzung $\mathcal{M}^*$ von $\mathcal{M}$ finden mit $\mathcal{M}^* \models A_0 \wedge A_1 \wedge A_2$. Dazu ist nur das Prädikatszeichens p in $\mathcal{M}^*$ zu interpretieren. Für ein beliebiges n-Tupel $\bar{a}$=$(a_1, \dots, a_n)$ von Elementen aus M setzten wir

$$\begin{array}{lll}
\mathcal{M}^* \models p[a_1, \dots a_n] & falls & a_i = \mathcal{M}(t_i^1) \text{ falls } t_i^1 \text{ eine Konstante ist und} \\
 & & \mathcal{M} \models \neg K_1[a_1, \dots, a_n] \\
\mathcal{M}^* \models \neg p[a_1, \dots a_n] & falls & \text{es } b_1, \dots, b_n \text{ gibt mit} \\
 & & a_i = \mathcal{M}(t_i^2)[b_1, \dots, b_n] \text{ für alle } i \text{ und} \\
 & & \mathcal{M} \models \neg K_2[b_1, \dots, b_n] \\
\text{beliebig} & sonst &
\end{array}$$

Damit gilt offensichtlich $\mathcal{M}^* \models \forall \bar{y}(K_1 \vee p(t_1^1, \dots t_n^1))$ und $\mathcal{M}^* \models \forall \bar{z}(K_2 \vee \neg p(t_1^1, \dots t_n^1))$. Die Schwierigkeit liegt aber woanders. Was geschieht für ein

n-Tupel $\bar{a}=(a_1, \ldots, a_n)$ das die ersten beiden Bedingungen in der Definition von p simultan erfüllt? Wir argumentieren, daß dieser Fall nicht vorkommen kann. Man sieht nämlich leicht, daß in diesem Fall nicht nur $\mathcal{M} \models \neg K_1[a_1, \ldots, a_n]$ und $\mathcal{M} \models \neg K_2[a_1, \ldots, a_n]$ gelten, sondern sogar $\mathcal{M} \models \sigma(\neg K_1 \wedge \neg K_2)[a_1, \ldots, a_n]$. Das ist aber ein Widerspruch zur Voraussetzung $\mathcal{M} \models \forall \bar{w} \sigma(K_1 \vee K_2)$.

■

Übersetzung in Logik zweiter Stufe Wir übersetzen die modale Formel A in eine Formel A^2 zweiter Stufe, so daß für jeden Kripke Rahmen (G, R) gilt:

$$\text{für alle } v \text{ gilt } (G, R, v) \models A \text{ gdw } (G, R) \models A^2$$

Siehe Abschnitt 3.6.2 für weitere Einzelheiten.

Formulierung des Problem Gegeben sei eine modale Formel A. Gesucht ist eine Formel F der Logik erster Stufe, so daß

$$A^2 \leftrightarrow F$$

allgemeingültig ist. Allgemeingültigkeit ist hier im Sinne der Logik zweiter Stufe zu verstehen.

Es stellt sich heraus, daß es günstiger ist eine Formel G in der Logik erster Stufe zu finden, so daß

$$\neg A^2 \leftrightarrow G$$

allgemeingültig ist. Das gewünschte F ist dann natürlich $\neg G$.

Quantoren verschwinden Es ist jetzt an der Zeit von der besonderen Form der Formel A^2 Gebrauch zu machen. Es handelt sich dabei um eine sehr harmlose Sorte von Formeln der Logik zweiter Stufe. A^2 hat die Form $\forall X_1, \ldots, \forall X_k A_0$, wobei A_0 keine weiteren Quantoren zweiter Stufe mehr enthält. $\neg A^2$ ist also äquivalent zu $\exists X_1, \ldots, \exists X_k \neg A_0$. Sei jetzt A_1 die Formel einer Logik erster Stufe, die aus A_0 entsteht, indem alle Vorkommen der Mengenvariablen X_i durch neue einstellige Prädikatssymbole p_i ersetzt werden. Nach Lemma 44 sind $\exists X_1, \ldots, \exists X_k \neg A_0$ und $\neg A_1$ se-äquivalent.

Umformungen Jetzt kann man mit allen die Erfüllbarkeit strikt erhaltenden Methoden, die aus dem automatischen Beweisen bekannt sind, versuchen, von $\neg A_1$ ausgehend eine Formel zu finden, in der die Prädikate p_i nicht mehr vorkommen.

Beispiel Wir betrachten den einfachsten Fall

$$A = \Box p \rightarrow p.$$

Damit

$$A^2 = \forall X \forall x (\forall y (R(x,y) \rightarrow X(y)) \rightarrow X(x)).$$

Also

$$\neg A^2 = \exists X \exists x (\forall y (R(x,y) \rightarrow X(y)) \wedge \neg X(x)).$$

Wir suchen also eine se-äquivalente Formel zu

$$\exists x (\forall y (R(x,y) \rightarrow p(y)) \wedge \neg p(x)),$$

in der p nicht mehr vorkommt. Versuchen wir es mit dem Resolutionskalkül. Skolemisierung liefert

$$(\forall y (R(a,y) \rightarrow p(y)) \wedge \neg p(a)).$$

Übergang zu Klauselnormalform liefert die beiden Klauseln

$$\forall y (\neg R(a,y) \vee p(y)) \qquad \neg p(a).$$

Die Resolution beider Klauseln liefert $\neg R(a,a)$ und somit die se-äquivalente Klauselmenge

$$\forall y (\neg R(a,y) \vee p(y)) \quad \neg p(a) \quad \neg R(a,a)$$

Man sieht aber leicht, daß diese se-äquivalent ist zu

$$\neg R(a,a)$$

alleine. Diese Formel ist ihrerseits se-äquivalent zu $\exists x \neg R(x,x)$. Die gesuchte charakterisierende Formel zu $\Box p \rightarrow p$ ist jetzt die Negation der bisher gefundenen Formel, d.h.

$$\forall x R(x,x).$$

Zweites Beispiel Wir betrachten

$$B = \Box p \to \Box\Box p$$

Also

$$B^2 = \forall X \forall x((\forall y(R(x,y) \to X(y))) \to \forall z(R(x,z) \to \forall w(R(z,w) \to X(w))))$$

und

$$\neg B^2 = \exists X \exists x((\forall y(R(x,y) \to X(y))) \land \exists z(R(x,z) \land \exists w(R(z,w) \land \neg X(w))))$$

Skolemisierung liefert die Klauselmenge

$$(\forall y(\neg R(a,y) \lor p(y))) \quad R(a,b) \quad R(b,c) \quad \neg p(c)$$

Resolution mit der Klausel $\neg p(c)$ bringt uns zu

$$(\forall y(\neg R(a,y) \lor p(y))) \quad R(a,b) \quad R(b,c) \quad \neg p(c) \quad \neg R(a,c)$$

Diese Klauselmenge ist se-äquivalent zu

$$R(a,b) \quad R(b,c) \quad \neg R(a,c)$$

Nach Lemma 44 ist dazu

$$\exists x \exists y \exists z(R(x,y) \land R(y,z) \land \neg R(x,z))$$

se-äquivalent. Die Negation davon liefert erwartungsgemäß

$$\forall x \forall y \forall z(R(x,y) \land R(y,z) \to R(x,z))$$

Ein interessantes Projekt wurde von Hans-Jürgen Ohlbach am Max–Planck–Institut in Saarbrücken realisiert. Er übersetzt beliebige Formeln der modalen Aussagenlogik in prädikatenlogische Klauselnormalform und versucht durch geschickte Anwendung des Resolutionskalküls die aussagenlogischen Variablen, auf der prädikatenlogischen Ebene durch einstellige Relationen repräsentiert, zu eliminieren. Gelingt das, so ist ein Charakterisierungsresultat in dem oben definierten Sinn bewiesen. Das Programm heißt SCAN und kann auch für eine Reihe anderer Aufgabenstellungen in auch anderen nichtklassischen Logiken benutzt werden. Man kann das Programm direkt über www aufrufen unter

`http:/www.mpi-sb.mpg.de/guide/staff/ohlbach/scan`

3.2.2 Übungsaufgaben

Übungsaufgabe 3.2.1 *Beweisen Sie die Resultate aus Abbildung 3.5.*

Übungsaufgabe 3.2.2 *Finden Sie eine Charakterisierende modale Formel für quasi-totale Rahmen (G, R), d.h. Rahmen die $(G, R) \models \forall x \forall y \forall z (R(x, y) \wedge R(x, z) \rightarrow (R(y, x) \vee R(x, y))$.*

Übungsaufgabe 3.2.3 *Für jede natürliche Zahl n, jede Kripke Struktur $\mathcal{K} = (G, R, v)$ und jede Welt $g \in G$ gilt:*

1. $(\mathcal{K}, g) \models \Box^n F$ *gdw für alle* $h \in G$ *mit* $R^n(g, h)$ *gilt* $(\mathcal{K}, h) \models F$
2. $(\mathcal{K}, g) \models \Diamond^n F$ *gdw es gibt* $h \in G$ *mit* $R^n(g, h)$ *und* $(\mathcal{K}, h) \models F$

Übungsaufgabe 3.2.4 *Sei (G, R) ein Rahmen. Dann gelten die folgenden Äquivalenzen:*

1	$(G, R) \models C(0, 1, 2, 0))$	*gdw*	*R ist transitiv*
2	$(G, R) \models C(0, 1, 0, 0))$	*gdw*	*R ist reflexiv*
3	$(G, R) \models C(1, 1, 0, 0))$	*gdw*	*R ist symmetrisch*
4	$(G, R) \models C(1, 0, 1, 0))$	*gdw*	*R ist funktional*

Siehe [Sterling, 1992], Seite 493.

Übungsaufgabe 3.2.5 *Zeigen Sie, daß die durch*

$$\Box(p \vee q) \rightarrow (\Box p) \vee (\Box q)$$

charakterisierte Klasse $\mathcal{R}$ genau aus den Rahmen (G, R) besteht die

$$\forall g \forall g_1 \forall g_2 (R(g, g_1) \wedge R(g, g_2) \rightarrow g_1 = g_2)$$

Solche Rahmen nennt man deterministisch.

Vergleichen Sie auch mit Aufgabe 3.1.2.

Übungsaufgabe 3.2.6 *Welche Klasse von Rahmen wird durch*

$$\neg \Box p \rightarrow \Box \neg p$$

charakterisiert?

Übungsaufgabe 3.2.7 *Ein Rahmen* (G_2, R_2) *heißt ein* **Unterrahmen** *eines Rahmens* (G_1, R_1)*, wenn*

1. $G_2 \subseteq G_1$ *und*
2. *für alle* $g, h \in G_2$ *gilt* $R_2(g, h)$ *gdw* $R_1(g, h)$

Ein Unterrahmen (G_2, R_2) *von* (G_1, R_1) *heißt ein* **abgeschlossener** *Unterrahmen, wenn jede Welt, die aus* G_2 *erreichbar ist, auch zu* G_2 *gehört. Formal:*

3. *für alle* $g, h \in G_1$ *mit* $g \in G_2$ *und* $R_1(g, h)$ *gilt* $h \in G_2$.

Zeigen Sie:
Ist (G_2, R_2) *ein abgeschlossener Unterrahmen von* (G_1, R_1)*, dann gilt für jedes* v_1*, jede Welt* $g \in G_2$ *und jede Formel in* Fml_{ALmod}

$$(G_1, R_1, v_1, g) \models F \text{ gdw } (G_2, R_2, v_2, g) \models F.$$

Hierbei ist v_2 *die Einschränkung der Funktion* v_1.

Übungsaufgabe 3.2.8 *Zeigen Sie mit Hilfe des Resultats aus Aufgabe 3.2.7, daß die Klasse aller nicht leeren Rahmen, d.h. die Klasse aller Rahmen* (G, R) *mit* $\exists x \exists y R(x, y)$ *nicht durch eine Formel der modalen Aussagenlogik charakterisiert werden kann.*

Übungsaufgabe 3.2.9 *Für zwei Rahmen* (G_1, R_1), (G_2, R_2) *besteht die* **disjunkte Summe** $(G, R) = (G_1, R_1) \uplus (G_2, R_2)$ *aus der disjunkten Summe* $G = G_1 \uplus G_2$ *der zugehörigen Welten. Falls* G_1 *und* G_2 *disjunkt sind ist also* G *die normale mengentheoretische Vereinigung* $G_1 \cup G_2$*. Anderenfalls müssen* G_1, G_2 *zuerst disjunkt gemacht werden: etwa* $G = (\{1\} \times G_1) \cup (\{2\} \times G_2)$*. Wir nehmen für die Zwecke dieser Übungsaufgabe an, daß* G_1 *und* G_2 *disjunkt sind und überlassen die Formulierung des allgemeinen Falles dem Leser. Die Relation* R *auf* G *ist relationentheoretisch ebenfalls die Summe von* R_1 *und* R_2*, d.h.*

$$R(g, h) \text{ gdw } R_1(g, h) \text{ oder } R_2(g, h)$$

Sind Belegungen v_1 *und* v_2 *auf* (G_1, R_1) *und* (G_2, R_2) *gegeben, dann ist wegen der Disjunktheit die folgende Definition möglich:*

$$v(g,p) = \begin{cases} v_1(g,p) & \textit{falls } g \in G_1 \\ v_2(g,p) & \textit{falls } g \in G_2 \end{cases}$$

Die Kripke Struktur $\mathcal{K} = (G, R, v)$ *nennen wir die disjunkte Summe von* $\mathcal{K}_1 = (G_1, R_1, v_1)$ *und* $\mathcal{K}_2 = (G_2, R_2, v_2)$.
Zeigen Sie für jede Formel $F \in Fml_{ALmod}$*:*

$$\mathcal{K} \models F \textit{ gdw } \mathcal{K}_1 \models F \textit{ oder } \mathcal{K}_2 \models F$$

Übungsaufgabe 3.2.10 *Zeigen Sie mit Hilfe von Aufgabe 3.2.9, daß die Klasse aller universellen Rahmen, d.h. die Klasse aller Rahmen* (G, R) *mit* $\forall x \forall y R(x, y)$ *nicht durch eine Formel der modalen Aussagenlogik charakterisiert werden kann.*

Übungsaufgabe 3.2.11 *Sei* (G, R_a, R_b) *ein multimodaler Kripke Rahmen, so daß* R_a *die transitive Hülle von* R_b *ist. Zeigen Sie, daß für jede Belegungsfunktion* v*, die folgenden Formeln in der Kripke Struktur* $\mathcal{K} = (G, R_a, R_b, v)$ *gültig sind:*

1. $\Box_a p \rightarrow (p \wedge \Box_a \Box_b p)$
2. $(\Box_a (p \rightarrow \Box_b p) \rightarrow (p \rightarrow \Box_a p))$
3. $\Box_a p \rightarrow (p \wedge \Box_b \Box_a q p)$

Übungsaufgabe 3.2.12 *Beweisen Sie Lemma 41*

Übungsaufgabe 3.2.13 *Sei* $\mathcal{R}$ *eine Klasse von Kripke Rahmen und seien* A_1*,* A_2 *modale Formeln, die beide* $\mathcal{R}$ *charakterisieren. Sind dann* A_1 *und* A_2 *notwendigerweise logisch äquivalent?*

Übungsaufgabe 3.2.14 *Berechnen Sie nach dem im Abschnitt 3.2.1 beschriebenen Konstruktionsverfahren charakterisierende Formeln der Prädikatenlogik erster Stufe für die folgenden modalen Formeln:*

1. $\Box p \rightarrow \Diamond p$
2. $p \rightarrow \Box \Diamond p$

3. $\Diamond^m\Box^n p \rightarrow \Box^j\Diamond^k p$

Übungsaufgabe 3.2.15 *Nach den Resultaten in Abbildung 3.5 kann man die Klasse der* Quasiordnungen*, d.h. die Klasse der reflexiven und transitiven Relationen, modal charakterisieren. Um eine partielle Ordnung zu erhalten müsste man noch die Eigenschaft der* Antisymmetrie *fordern, d.h.* $\forall x\forall y(R(x,y) \wedge R(y,x) \rightarrow x = y)$

Zeigen Sie, daß die Klasse der antisymmetrischen Rahmen nicht charakterisierbar ist.

Übungsaufgabe 3.2.16 *In Lemma 44 (5) wird verlangt, daß die Argumentterme* $t_1^1, \ldots, t_n^1$ *von p Variablen oder Konstanten sein müssen. Ist diese Einschränkung verzichtbar?*

Dazu betrachten wir die Klauselmenge S:

$$r(f(c)) \quad q(a) \quad \neg q(b) \quad \forall x(q(x) \vee p(f(x))) \quad \neg p(f(a))$$

Unter Verletzung der in Lemma 44 (5) geforderten Einschränkung erhält man als Resolvente der letzten beiden Klauseln

$$q(a)$$

1. *Zeigen Sie, daß die Klauselmenge* $\{r(f(c)) \quad q(a) \quad \neg q(b)\}$ *nicht se-äquivalent ist zu* $\{r(f(c)) \quad q(a) \quad \neg q(b) \quad \forall x(q(x) \vee p(f(x))) \quad \neg p(f(a))\}$
2. *Mit* S^* *bezeichen wir die zu S äquivalente Klauselmenge*

   $$r(f(c)) \quad q(a) \quad \neg q(b) \quad \forall x\forall u(q(x) \vee u \neq f(x) \vee p(u)) \quad \neg p(f(a))$$

   Berechnen Sie nach Lemma 44 (5) eine zu S^* *se-äquivalente Klauselmenge, die p. nicht mehr enthält.*

Übungsaufgabe 3.2.17 *Ist die Klasse aller Rahmen* (G, R)*, wobei R die Gleichheitsrelations auf G ist charakterisierbar?*

Geben Sie eine charakterisierende Formel an oder eine Beweis der Nichtcharakterisierbarkeit.

Übungsaufgabe 3.2.18 *Ist die folgende Vermutung*

> *Seien ϕ und ψ zwei Formeln der modalen Aussagenlogik, welche dieselbe Klasse von Rahmen $\mathcal{R}$ charakterisieren. Dann ist sind ϕ und ψ äquivalent, d.h. $\phi \leftrightarrow \psi$ ist eine Tautologie.*

wahr? Begründen Sie Ihre Antwort.

Übungsaufgabe 3.2.19 *Im Lemma 39 wird behauptet, daß für jede Kripke Struktur $\mathcal{K}$ und jede modale Formel F gilt*

$$\mathcal{K} \models F \Leftrightarrow \mathcal{K}^* \models F$$

Beginnt man mit einer Kripke Struktur $\mathcal{K}$ mit reflexivem Rahmen, so wird $\mathcal{K}^*$ sicher nicht reflexiv sein. Die Formel $\Box p \to p$ charakterisiert aber gerade reflexive Rahmen. Ist das nicht ein Widerspruch?
Zeigen Sie den Irrtum in dieser Argumentation auf.

3.3 Vollständigkeit und kanonische Modelle

Eine **modale Logik L** ist zunächst nichts weiter als eine Teilmenge von Fml_{ALmod}, wobei wir uns vorstellen, daß **L** die Menge aller Tautologien festlegt. Mit diesem Begriff allein läßt sich noch nicht viel anfangen, wir brauchen zusätzliche Eigenschaften von **L** um überhaupt sinnvolle Aussagen über modale Logiken beweisen zu können. Auf der anderen Seite sollen die Einschränkungen an **L** nicht zu stark sein, damit möglichst viele der bekannten Systeme der modalen Aussagenlogik darunterfallen. Es gibt tatsächlich eine Klasse modaler Logiken, die beide Wünsche erfüllt.

Definition 49
Eine Teilmenge **L** *von* Fml_{ALmod} *heißt eine* **normale Modallogik**, *wenn sie die folgenden Bedingungen erfüllt:*

1. **L** *enthält alle aussagenlogischen Tautologien.*
2. $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q) \in$ **L** *für zwei AL-Variable* p, q.
3. **L** *ist abgeschlossen unter modus ponens,*
   d.h. mit $A \in$**L** *und* $(A \rightarrow B) \in$**L** *gilt auch* $B \in$**L**.
4. *aus* $A \in$**L** *folgt* $\Box A \in$**L**.
5. **L** *ist abgeschlossen unter Instanziierungen,*
   d.h. gilt $A \in$ **L** *und ist* A' *eine Instanz von* A, *dann gilt auch* $A' \in$ **L**.

Dabei entsteht eine Instanz A' *aus* A, *indem für einige aussagenlogische Variablen* $p_1, \ldots, p_k$ *und beliebige Formeln* $B_1, \ldots, B_k$ *aus* Fml_{ALmod} *simultan jedes Vorkommen von* p_i *in* A *ersetzt wird durch* B_i.

Wir sind aussschließlich an modalen Logiken **L** interessiert, die aus der Menge aller Tautologien eines gegebenen modallogischen Systems **S** von Axiomen und Regeln bestehen.

Definition 50 *Ein modallogisches System* **S** *heißt ein* **normales modallogisches System**, *wenn*

1. *in* **S** *alle aussagenlogischen Tautologien abgeleitet werden können,*

2. $\Box(p \to q) \to (\Box p \to \Box q)$ *ein Axiom von* **S** *ist,*

3. *Modus Ponens und die Generalisierung zu den Regeln von* **S** *gehören.*

Mit Ausnahme von **C** sind alle modallogischen Systeme aus Abbildung 3.2 normal. Es ist außerdem offensichtlich, daß für jedes normale modallogisches System **S** die Menge $\{A \mid \vdash^{\mathbf{S}} A\}$ eine normale Modallogik ist.

Das übergeordnete Ziel in diesem Unterkapitel sind die Vollständigkeitssätze für modallogische Systeme. In einem ersten Teil werden wir zunächst Vorbereitungen dafür bereitstellen, die für jede normale Modallogik gelten.

Lemma 45 *Für eine normale Modallogik* **L** *gilt für beliebige Formeln A,B,C in* Fml_{ALmod}*:*

1. $\Box(A \wedge B) \leftrightarrow (\Box A \wedge \Box B) \in \mathbf{L}$

2. *Aus* $(A \to B) \in \mathbf{L}$ *folgt auch* $(\Box A \to \Box B) \in \mathbf{L}$

3. *Gilt* $(A \leftrightarrow B) \in \mathbf{L}$ *und* D *entsteht aus der Formel* C*, indem die Teilformel* A *in* C *durch* B *ersetzt wird, dann gilt auch* $(C \leftrightarrow D) \in \mathbf{L}$

Beweis: Übungsaufgabe 3.3.1

■

Definition 51
Sei **L** *eine normale Modallogik. Eine Teilmenge* $F \subseteq Fml_{ALmod}$ *heißt* **L-inkonsistent***, wenn es Formeln* $A_1, \dots, A_k \in F$ *gibt mit*

$$(\neg A_1 \vee \ldots \vee \neg A_k) \in \mathbf{L}.$$

F *heißt* **L-konsistent***, wenn* F *nicht* **L***-inkonsistent ist.*

Wenn aus dem Zusammenhang klar ist, welche Logik **L** gemeint ist oder wenn die Wahl einer speziellen Logik **L** keine Rolle spielt, so sprechen wir kurz von inkonsistenten bzw. konsistenten Formelmengen.

In der Definition einer normalen modalen Logik **L** wurde nicht verlangt, daß **L** konsistent sein muß, und in der Tat sind die meisten Lemmata dieses Abschnitts auch für inkonsistente Logiken wahr, wenn auch auf triviale Weise.

Wir wollen aber von jetzt an annehmen, daß alle normalen modalen Logiken **L** auch **L**-konsistent sind.

Definition 52 *Eine konsistente Teilmenge* $F \subseteq Fml_{ALmod}$ *heißt* **maximal konsistent**, *wenn für jede Formel* $A \in Fml_{ALmod}$ *entweder* $A \in F$ *oder* $\neg A \in F$ *gilt.*

Lemma 46 *Sei* **L** *eine normale Modallogik und* $F \subseteq Fml_{ALmod}$ *eine maximal konsistente Teilmenge bezüglich* **L**. *Dann gelten die folgenden Aussagen:*

1. *für jede Formel* A *ist genau eine der beiden Formeln* A , $\neg A$ *Element von* F,
2. $(A \vee B) \in F$ *genau dann, wenn* $A \in F$ *oder* $B \in F$,
3. $(A \wedge B) \in F$ *genau dann, wenn* $A \in F$ *und* $B \in F$,
4. $\mathbf{L} \subseteq F$,
5. F *ist abgeschlossen unter modus ponens.*

Beweis:
zu (1) Daß eine der beiden Formeln A,$\neg A$ in F liegen muß ist gerade die Definition maximal konsistenter Formelmengen. Sind beide Formeln Elemente von F, so folgt daraus die Inkonsistenz von F,da $\neg A \vee \neg\neg A$ als AL-Tautologie ebenfalls in **L** liegt.
zu (2) Angenommen $(A \vee B) \in F$ aber $A \notin F$ und $B \notin F$. Nach Definition einer maximal konsistenten Menge gilt dann $\neg A \in F$ und $\neg B \in F$. Außerdem gilt $(\neg\neg A \vee \neg\neg B \vee \neg(A \vee B)) \in \mathbf{L}$, weil diese Formel eine AL-Tautologie ist. Daraus folgt aber ein Widerspruch zur Konsistenz von F. Damit ist die Implikation von links nach rechts bewiesen.
Gilt etwa $A \in F$ und $(A \vee B) \notin F$, dann folgt wieder aus der Maximalität von F, daß $\neg(A \vee B) \in F$ gilt. Als AL-Tautologie ist auch $\neg A \vee (A \vee B)$ ein Element von **L**, was wieder im Widerspruch zur Konsistenz von F steht.
zu (3) analog zu (2).
zu (4) Für $A \in \mathbf{L}$ ist $\{\neg A\}$ eine **L**-inkonsistente Menge. Also $\neg A \notin F$, dann muß aber $A \in F$ gelten.
zu (5) Aus der Annahme $A \in F$, $(A \to B) \in F$ und $B \notin F$,

folgt $\neg B \in F$ und daraus ein Widerspruch zur Konsistenz von F, weil $\neg A \vee \neg(A \rightarrow B) \vee B$ eine AL-Tautologie ist.
Man beachte, daß in diesem Beweis nur von der Tatsache Gebrauch gemacht wurde, daß $\mathbf{L}$ alle AL-Tautologien enthält.

∎

Lemma 47 *Zu jeder konsistenten Formelmenge F gibt es eine maximal konsistente Obermenge F_{max}.*

Beweis:
Wir fixieren für diesen Beweis eine Aufzählung $A_0, \ldots, A_n, \ldots$ aller Formeln aus Fml_{ALmod}. Wir definieren induktiv eine aufsteigende Kette von Formelmengen $F_0 \subseteq F_1 \ldots F_n \ldots$ durch:

- $F_0 = F$
- $F_{n+1} = \begin{cases} F_n \cup \{A_n\} & \text{falls diese Menge konsistent ist} \\ F_n \cup \{\neg A_n\} & \text{sonst} \end{cases}$

Wir zeigen, durch Induktion über n, daß F_n konsistent ist. Für $n = 0$ können wir auf die Voraussetzung über F zurückgreifen. Zum Beweis des Induktionsschrittes nehmen wir an, daß F_{n+1} inkonsistent ist. Nach Definition von F_{n+1} sind dann die beiden Mengen $F_n \cup \{A_n\}$ und $F_n \cup \{\neg A_n\}$ inkonsistent. Dann gibt es nach Definition der Inkonsistenz Formeln $B_1, \ldots, B_k, C_1, \ldots C_r \in F_n$, so daß:

$$\neg B_1 \vee \ldots \vee \neg B_k \vee \neg A_n \in \mathbf{L} \tag{1}$$

und

$$\neg C_1 \vee \ldots \vee \neg C_r \vee A_n \in \mathbf{L} \tag{2}$$

Da $\mathbf{L}$ alle AL-Tautologien enthält und unter modus ponens abgeschlossen ist folgt aus (1) und (2) auch

$$\neg B_1 \vee \ldots \vee \neg B_k \vee \neg C_1 \vee \ldots \vee \neg C_r \in \mathbf{L}$$

im Widerspruch zur Induktionsvoraussetzung, daß F_n konsistent ist.
Setzt man die Vereinigung aller F_n gleich F_{max}, dann zeigt man leicht

1. F_{max} ist konsistent
2. F_{max} ist maximal konsistent
3. $F \subseteq F_{max}$

Man beachte, daß in diesem Beweis nur benutzt wurde, daß **L** alle AL-Tautologien enthält und unter modus ponens abgeschlossen ist.

∎

Jetzt kommen die modalen Operatoren ins Spiel.

Definition 53 *Für jede Menge $F \subseteq Fml_{ALmod}$ setzen wir:*

$\Box^- F =$	$\{A \mid \Box A \in F\}$	$\Box^+ F =$	$\{\Box A \mid A \in F\}$
$\Diamond^- F =$	$\{A \mid \Diamond A \in F\}$	$\Diamond^+ F =$	$\{\Diamond A \mid A \in F\}$

Lemma 48 *Sind F konsistent und gilt $\neg\Box A \in F$, dann ist auch $\Box^- F \cup \{\neg A\}$ konsistent.*

Beweis: Wäre $\Box^- F \cup \{\neg A\}$ inkonsistent, dann gäbe es Formeln $C_1, \ldots, C_k$ in $\Box^- F$, so daß

$$(\neg C_1 \vee \ldots \vee \neg C_k \vee A) \in \mathbf{L} \tag{1}$$

Zunächst ist nicht klar, daß A selbst in der Disjunktion der negierten Formeln aus $\Box^- F \cup \{\neg A\}$, welche die Inkonsistenz belegen, vorkommt. Aus $G \in \mathbf{L}$ folgt aber umso mehr $G \vee G_1 \in \mathbf{L}$ für beliebige Formeln G, G_1. Also können wir in jedem Fall (1) annehmen.

Mit aussagenlogischen Argumenten allein folgt daraus

$$(C_1 \wedge \ldots \wedge C_k \to A) \in \mathbf{L} \tag{2}$$

mit Lemma 45(2) folgt

$$(\Box(C_1 \wedge \ldots \wedge C_k) \to \Box A) \in \mathbf{L} \tag{3}$$

mit Lemma 45(1) und (3) folgt

$$(\Box C_1 \wedge \ldots \wedge \Box C_k) \to \Box A) \in \mathbf{L} \tag{4}$$

woraus wieder allein mit Aussagenlogik

$$(\neg\Box C_1 \vee \ldots \vee \neg\Box C_k \vee \Box A) \in \mathbf{L} \tag{5}$$

erhalten werden kann. Aus (5) folgt unmittelbar ein Widerspruch zur Konsistenz von F.

∎

Bemerkung: Wenn F eine konsistente Formelmenge ist, dann muß $\Box^- F$ nicht notwendigerweise auch konsistent sein. Ist zum Beispiel $\mathcal{K} = (\{g\}, \emptyset, v)$ eine Kripke Struktur mit nur einer einzigen Welt und einer leeren

Zugänglichkeitsrelation, dann ist $F = \{A \mid (\mathcal{K}, g) \models A\}$ eine maximal konsistente Formelmenge bezüglich der minimalen nomalen Logik **K**. Da F für jede beliebige Formel A sowohl $\Box A$ als auch $\Box\neg A$ enthält ist $\Box^- F$ inkonsistent. Unter der Voraussetzung des vorangegangen Lemmas aber, $\neg\Box A \in F$, folgt aus der Konsistenz von F auch die Konsistenz von $\Box^- F$ konsistent

Wir weisen darauf hin, daß zum Beweis dieses Lemmas wesentlich stärker von der Definition einer normalen modalen Logik Gebrauch gemacht wurde, nämlich von den Klauseln (2), (4) und (5) aus Definition 49 auf dem Umweg über Lemma 45.

Definition 54 *Zu einer normalen Modallogik* **L** *ist die* **kanonische Kripke Struktur** $\mathcal{K}_{\mathbf{L}} = (G_{\mathbf{L}}, R_{\mathbf{L}}, v_{\mathbf{L}})$ *gegeben durch:*

- $G_{\mathbf{L}}$ *ist die Menge aller maximal konsistenten Formelmengen bezüglich* **L**,
- *für* $W_1, W_2 \in G_{\mathbf{L}}$ *gilt* $R_{\mathbf{L}}(W_1, W_2)$ *genau dann, wenn* $\Box^- W_1 \subseteq W_2$,
- *für jede AL-Variable* p *und* $W \in G_{\mathbf{L}}$ *setzt man* $v_{\mathbf{L}}(W, p) = 1$ *genau dann, wenn* $p \in W$.

Satz 49 *Sei* **L** *eine normale Modallogik und* $\mathcal{K}_{\mathbf{L}} = (G_{\mathbf{L}}, R_{\mathbf{L}}, v_{\mathbf{L}})$ *ihre kanonische Kripke Struktur. Dann gilt für jede Welt* $W \in G_{\mathbf{L}}$ *und jede Formel* $A \in Fml_{ALmod}$:

$$(\mathcal{K}_L, W) \models A \text{ gdw } A \in W$$

Beweis: Durch Induktion über den Aufbau der Formel A.
Der Induktionsanfang, A ist eine AL-Variable, folgt unmittelbar aus der Definition der kanonischen Kripke Struktur.
Wir bringen zwei Beispiele für die Beweise im Induktionsschritt.

$A = A_1 \wedge A_2$ Unter Ausnutzung der Eigenschaften maximal konsistenter Formelmengen, siehe Lemma 46, und der Induktionsvoraussetzung ergibt sich die folgende Argumentationskette:

$$\begin{array}{lll} (\mathcal{K}_L, W) \models (A_1 \wedge A_2) & \text{gdw} & (\mathcal{K}_L, W) \models A_1 \text{ und } (\mathcal{K}_L, W) \models A_2 \\ & \text{gdw} & A_1 \in W \text{ und } A_2 \in W \\ & \text{gdw} & (A_1 \wedge A_2) \in W \end{array}$$

$A = \Box B$ Wir nehmen zunächst $\Box B \in W$ an und betrachten eine Welt W_1 mit $R_{\mathbf{L}}(W, W_1)$. Nach Definition von $R_{\mathbf{L}}$ gilt $B \in W_1$ und daher nach Induktionsvoraussetzung $(\mathcal{K}_L, W_1) \models B$. Damit ist aber $(\mathcal{K}_L, W) \models \Box B$.

Gelte jetzt $\Box B \not\in W$, d.h. $\neg\Box B \in W$. Nach Lemma 48 ist dann die Menge $\Box^- W \cup \{\neg B\}$ konsistent und nach Lemma 47 gibt es eine Welt $W_1 \in G_{\mathbf{L}}$ mit $\Box^- W \cup \{\neg B\} \subseteq W_1$. Nach Definition von $R_{\mathbf{L}}$ gilt $R_{\mathbf{L}}(W, W_1)$ und nach Induktionsvoraussetzung $(\mathcal{K}_L, W_1) \models \neg B$. Insgesamt erhalten wir $(\mathcal{K}_L, W) \models \neg\Box B$.

$\blacksquare$

Korollar 50 *Ist $\mathcal{K}_{\mathbf{L}}$ die kanonische Kripke Struktur der normale Modallogik $\mathbf{L}$, dann gilt für jede Formel $A \in Fml_{ALmod}$:*

$$A \in \mathbf{L} \textit{ genau dann, wenn } \mathcal{K}_L \models A$$

Beweis:
Nach Lemma 46 ist $\mathbf{L}$ eine Teilmenge jeder Welt $W \in G_{\mathbf{L}}$. Die Implikation von links nach rechts folgt direkt aus dem Satz 49.

Gelte jetzt umgekehrt $\mathcal{K}_L \models A$, d.h. für jede Welt W in der kanonischen Kripkestruktur gilt $(\mathcal{K}_L, W) \models A$. Angenommen es gilt $A \not\in \mathbf{L}$. Dann muß $\mathbf{L} \cup \{\neg A\}$ konsistent sein, denn anderenfalls gäbe es $C_1, \ldots, C_k \in \mathbf{L}$ mit $(\neg C_1 \vee \ldots \vee \neg C_k \vee A) \in \mathbf{L}$. i.e. $(C_1 \wedge \ldots \wedge C_k \rightarrow A) \in \mathbf{L}$. Wegen der Eigenschaften von $\mathbf{L}$ gilt dann $A \in \mathbf{L}$. Wenn $\mathbf{L} \cup \{\neg A\}$ aber konsistent ist, dann gibt es eine Welt $W' \in G_{\mathbf{L}}$ mit $\mathbf{L} \cup \{\neg A\} \subseteq W'$, i.e. $\neg A \in W'$. Aus Satz 49 folgt damit der Widerspruch $(\mathcal{K}_L, W') \not\models A$. Also war die Annahme $A \not\in \mathbf{L}$ falsch und es muß vielmehr $A \in \mathbf{L}$ gelten.

$\blacksquare$

Korollar 51 *Sei $\mathbf{S}$ ein normales modallogisches System und $\mathcal{K}_{\mathbf{L}}$ die kanonische Kripke Struktur der Logik $\mathbf{L} = \{A \mid \vdash^{\mathbf{S}} A\}$. Dann gilt*

$$\vdash^{\mathbf{S}} A \quad \textit{gdw} \quad \mathcal{K}_{\mathbf{L}} \models A$$

Beweis:
Setze $\mathbf{L} = \{A \mid \vdash^{\mathbf{S}} A\}$ im vorangegangenen Korollar 50.

■

Das Korollar 51 kann als ein Vollständigkeitsresultat aufgefasst werden, es stellt einen Zusammenhang her zwischen einem syntaktischen Ableitungsbegriff, $\vdash^S$, und einem semantischen Begriff, Gültigkeit im kanonischen Modell. Der aus der Logik erster Stufe bekannte Vollständigkeitssatz hat eine geringfügig andere Form; auf der semantischen Seite wird nicht wie in Korollar 51 auf eine einzige Struktur bezug genommen sondern auf die Klasse aller Strukturen. Wir wollen jetzt Sätze dieses Typs auch für Modallogiken formulieren. Zunächst betrachten wir das modallogische System **K**, siehe Abbildung 3.1. Offensichtlich ist **K** ein normales modallogisches System, so daß die Resultate des vorangegangenen Abschnitts zur Verfügung stehen. Alle Systeme, die wir in diesem Unterkapitel noch betrachten werden, sind normal.

Satz 52 (Vollständigkeit für K) *Für jede modallogische Formel A gilt*

$$\vdash^{\mathbf{K}} A \text{ gdw } \mathcal{M} \models A \text{ für jede Kripke Struktur } \mathcal{M}$$

Beweis:
Gilt $\mathcal{M} \models A$ für jede Kripke Struktur $\mathcal{M}$, dann gilt auch $\mathcal{K} \models A$ für das kanonische Kripke Modell und damit nach Korollar 51 auch $\vdash^{\mathbf{K}} A$.
Für die umgekehrte Implikation zeigt man, daß jedes Axiom von **K** in allen Kripke Modellen gilt und jede Regel von **K** Gültigkeit erhält. Der Nachweis der Allgemeingültigkeit der Axiome ist nur für das Axiom $\Box(A \to B) \to (\Box A \to \Box B)$ nicht sofort klar. Ist $\mathcal{M}$ eine beliebige Kripke Struktur und g eine beliebige Welt in $\mathcal{M}$, mit $(\mathcal{M}, g) \models \Box(A \to B)$ und $(\mathcal{M}, g) \models \Box A$, so müssen wir $(\mathcal{M}, g) \models \Box B$ zeigen, d.h. für jede Welt h mit $R(g, h)$ ist $(\mathcal{M}, h) \models B$ zu zeigen. Für jedes solche h gilt aber nach Voraussetzung $(\mathcal{M}, h) \models A \to B$ und $(\mathcal{M}, h) \models A$ und wir sind fertig.
Da hinlänglich bekannt ist, daß modus ponens Allgemeingültigkeit erhält, braucht nur noch die Regel (G) betrachtet zu werden. Gilt aber $\mathcal{M} \models A$, dann gilt nach Definition der Modellrelation $(\mathcal{M}, g) \models A$ für jede Welt g in $\mathcal{M}$. Um so mehr gilt dann für jedes fest gewählte g, daß für alle von g aus erreichenbaren Welten h auch $(\mathcal{M}, h) \models A$ gilt. Das heißt aber $(\mathcal{M}, g) \models \Box A$.

■

Satz 53 (Vollständigkeit für T) *Für jede modallogische Formel A gilt*

$$\vdash^{\mathbf{T}} A \text{ gdw } \mathcal{M} \models A \text{ für jede reflexive Kripke Struktur } \mathcal{M}$$

Beweis:
Der Beweis läuft nach dem Muster des Beweises von Satz 52. Zwei Behauptungen sind dabei neu zu zeigen

1. das kanonische Modell für **T** ist reflexiv,
2. das Axiom ist wahr in jeder reflexiven Struktur.

Zu 1 Für jede **T**-maximale Menge w ist $\Box^- w \subseteq w$ nachzuweisen. Wegen Lemma 46(4) ist jede Instanz des Axioms $\Box p \rightarrow p$ von **T** ein Element von w. Liegt also eine Formel $\Box A$ in w, so gilt wegen der Abgeschlossenheit von w unter modus ponens (Lemma 46(5)) auch $A \in w$.

Zu 2 muß noch ergänzt werden.

■

Satz 54 (Vollständigkeit für S4) *Für jede modallogische Formel A gilt*

$$\vdash^{\mathbf{S4}} A \text{ gdw } \mathcal{M} \models A \text{ für jede reflexiv transitive Kripke Struktur } \mathcal{M}$$

Beweis:
siehe z.B. [Hughes & Cresswell, 1984]

Satz 55 (Vollständigkeit für S5) *Für jede modallogische Formel A gilt*

$$\vdash^{\mathbf{S5}} A$$

gdw

$\mathcal{M} \models A$ für jede Kripke Struktur $\mathcal{M}$, deren Zugänglichkeitsrelations eine Äquivalenzrelation ist.

Beweis:
siehe z.B. [Hughes & Cresswell, 1984]

Definition 55 *Ist S ein modallogisches System und Σ eine Formelmenge, dann bedeutet*

$$\Sigma \vdash^{\mathbf{S}} F,$$

die Existenz eine Folge von Formeln $F_1, F_2, \ldots, F_k$, so daß

- *die letzte Formel ist F, d.h. $F_k = F$,*
- *jedes F_j für $1 \leq j \leq k$ ist*
  - *entweder Instanz eines Axioms von S oder*
  - *ein Element der Prämissenmenge Σ oder*
  - *folgt aus Formeln mit kleinerem Index mit Hilfe einer Regel des Systems S.*

Satz 56 (Starke Vollständigkeit) *Sei* **S** *eines der normalen, modallogischen Systeme aus den Sätzen 52, 53, 54, 55. Dann gilt für Menge Σ modallogischer Formeln und jede modallogische Formel A*

$$\Sigma \models_G^{\mathbf{S}} \Leftrightarrow \Sigma \vdash^{\mathbf{S}} A$$

Beweis: Die Richtung von rechts nach links, also die Korrektheitsbehauptung, erfordert keine ¨Uberlegung, die über die in den drei genannten Beweisen benutzte hinausgeht.

Als Vorbereitung für den zweiten Teil setzen wir $\mathbf{L} = \{A \mid \Sigma \vdash^{\mathbf{S}} A\}$. Es ist leicht zu sehen, daß $\mathbf{L}$ eine normale Modallogik ist. Sei $\mathcal{K} = \mathcal{K}_{\mathbf{L}}$ das kanonische zu $\mathbf{L}$. Zum Beweis der umgekehrten Richtung nehmen wir also an, daß $\Sigma \models_G^{\mathbf{S}}$ an. Wegen $\Sigma \subseteq \mathbf{L}$ gilt jedenfalls $\mathcal{K} \models \Sigma$ und damit auch $\mathcal{K} \models A$. Nach Korollar 50 und Definition von $\mathbf{L}$ gilt dann auch $\Sigma \vdash^{\mathbf{S}} A$.

■

3.3.1 Übungsaufgaben

Übungsaufgabe 3.3.1 *Beweisen Sie Lemma 45.*

Übungsaufgabe 3.3.2 *Zeigen Sie für eine normale Modallogik* **L***:*
F ist eine maximal konsistente Formelmenge bzgl. **L** *genau dann, wenn jede echte Obermenge von F* **L***-inkonsistent ist.*

Übungsaufgabe 3.3.3

Definition 56 *Sei $A \in Fml_{ALmod}$. Eine Formel von der Form $\Box^n A'$, wobei A' eine Instanz von A ist heißt eine* Boxinstanz *von A.*

*Die normale Modallogik **L** entstehe aus dem Basissystem **K** durch Hinzunahme der Axiom $A_1, \ldots, A_k$. Beweisen Sie die folgende modale Version des Deduktionstheorems:*

Theorem 57 (Deduktionstheorem)
Gilt für eine Formel $B \in Fml_{ALmod}$

$$\mathbf{L} \vdash B$$

dann gibt es Boxinstanzen $C_1, \ldots C_r$ der Axiome $A_1, \ldots, A_k$ mit

$$\mathbf{K} \vdash C_1 \wedge \ldots \wedge C_r \rightarrow B$$

Hinweis: Machen Sie davon Gebrauch, daß in der klassischen, nicht modalen Aussagenlogik des Deduktionstheorem gilt.

Übungsaufgabe 3.3.4 *Wir betrachten die beiden Formeln*

H	$\Box(p \vee q) \wedge \Box(\Box p \vee q) \wedge \Box(p \vee \Box q) \rightarrow (\Box p \vee \Box q)$
L	$\Box(p \wedge \Box p \rightarrow q) \vee \Box(p \wedge \Box q \rightarrow p)$

Zeigen Sie, daß beide Formeln äquivalent sind, d.h. daß $\mathbf{KH} \vdash L$ und $\mathbf{KL} \vdash H$ gilt. Siehe [Heuerding, 1998, Theorem 11.3.5].

Übungsaufgabe 3.3.5 *Nach vier Beispielen fragt man sich natürlich nach einem allgemeinen umfassenden Satz über Vollständigkeitsresultate. Hier ist eine naheliegende Vermutung:*

Vollständigkeitvermutung für charakterisierbare Klassen *Sei $\mathcal{R}$ eine Klasse von Rahmen, die durch die Formeln $C_1, \ldots, C_k \in Fml_{ALmod}$ charakterisiert wird und $\mathcal{K}$ die Klasse aller Kripke Strukturen mit Rahmen in $\mathcal{R}$. Das modallogische System S entstehe aus dem System **K** durch Hinzunahme von $C_1, \ldots, C_k$ als neue Axiome. Dann gilt für jedes $A \in Fml_{ALmod}$:*

$$\vdash^S A \text{ gdw } \mathcal{M} \models A \text{ für jede Kripke Struktur } \mathcal{M} \text{ in } \mathcal{K}$$

Unglücklicherweise ist diese Vermutung falsch. Näheres dazu findet man z.B. in [van Benthem, 2001].

1. *Zeigen Sie daß die Konjunktion der folgenden Formeln*

   - $\Box p \to p$
   - $\Box\Diamond p \to \Diamond\Box p$
   - $(\Diamond p \wedge \Box(p \to \Box p)) \to p$

   die Klasse aller Rahmen mit $\forall x \forall y (R(x,y) \leftrightarrow x = y)$ *charakterisiert*

2. *Zeigen Sie, daß die Formel* $\Box p \leftrightarrow p$ *diesselbe Klasse von Rahmen charakterisiert.*

3. *Zeigen Sie, daß* $\Box p \leftrightarrow p$ *keine logische Konsequenz aus dem System* **K** *mit den Formeln aus 1 als zusätzlichen Axiomen ist.*

3.4 Entscheidbarkeit modaler Aussagenlogiken

Definition 57 (Filtration)
Sei $\mathcal{K} = (G, R, v)$ eine Kripke Struktur und $\Gamma \subseteq Fml_{ALmod}$ eine Menge von Formeln der modalen Aussagenlogik. Die Äquivalenzrelation $\sim_\Gamma$ auf G wird definiert durch:

$$g \sim_\Gamma h \text{ gdw } (\mathcal{K}, g) \models A \Leftrightarrow (\mathcal{K}, h) \models A \text{ für alle } A \in \Gamma$$

Für $g \in G$ bezeichnen wir mit $[g]$ die Äquivalenzklasse von g bezüglich $\sim_\Gamma$, i.e. $[g] = \{h \in G \mid g \sim_\Gamma h\}$. Die Kripke Struktur $\mathcal{K}_\Gamma = (G_\Gamma, R_\Gamma, v_\Gamma)$ ist wie folgt definiert

$$\begin{array}{lcll}
G_\Gamma & = & \{[g] \mid g \in G\} & \\
{[g]} R_\Gamma [h] & \Leftrightarrow & \exists g_0 \in [g] \exists h_0 \in [h](g_0 R h_0) & \\
v_\Gamma([g], p) & = & v(g, p) & \text{für } p \in \Gamma \\
v_\Gamma([g], p) & = & \text{beliebig} & \text{sonst}
\end{array}$$

$\mathcal{K}_\Gamma$ heißt eine **Filtration** *von $\mathcal{K}$ bezüglich Γ.*

Lemma 58

1. *Sei $\Gamma \subseteq Fml_{ALmod}$ abgeschlossen unter Teilformeln, dann gilt für alle $A \in \Gamma$ und alle $g \in G$*

$$(\mathcal{K}, g) \models A \text{ gdw } (\mathcal{K}_\Gamma, [g]) \models A$$

2. *Ist Γ endlich mit $\#\Gamma = n$, dann ist $\#G_\Gamma \leq 2^n$.*

Beweis:
(1) Der Beweis geschieht durch Induktion über den Formelaufbau von A. Ist A eine Aussagenvariable, dann folgt die Behauptung aus der Definition von v_Γ und der Äquivalenzrelation $\sim_\Gamma$. Wir betrachten nur noch den Induktionsschritt von B auf $\Box B$. Gelte $(\mathcal{K}_\Gamma, [g]) \models \Box B$. Dann gilt für alle $h \in G$ mit $[g] R_\Gamma [h]$ auch $(\mathcal{K}_\Gamma, [h]) \models B$. Nach Induktionsvoraussetzung folgt damit für alle $h \in G$ mit $[g] R_\Gamma [h]$ auch $(\mathcal{K}, h) \models B$ und damit auch für alle $h \in G$ mit gRh ebenfalls $(\mathcal{K}, h) \models B$, also auch $(\mathcal{K}, g) \models \Box B$. Gelte jetzt umgekehrt

$(\mathcal{K}, g) \models \Box B$. Wir betrachten $g, h \in G$ mit $[g]R_\Gamma[h]$. Nach Definition gibt es $g_0, h_0 \in G$ mit $g_0 \sim_\Gamma g$, $h_0 \sim_\Gamma h$ und $g_0 R h_0$. Nach Definition von $\sim_\Gamma$ gilt auch $(\mathcal{K}, g_0) \models \Box B$ und damit auch $(\mathcal{K}, h_0) \models B$. Woraus mit der Definition von $\sim_\Gamma$ wieder $(\mathcal{K}, h) \models B$ folgt. Nach Induktionsvoraussetzung erhalten wir weiter $(\mathcal{K}_\Gamma, [h]) \models B$. Insgesamt ist damit für alle $[h] \in G_\Gamma$ mit $[g]R_\Gamma[h]$ die Gültigkeit von $(\mathcal{K}_\Gamma, [h]) \models B$ gezeigt und damit $(\mathcal{K}_\Gamma, [g]) \models \Box B$
(2) Enthält Γ n Formeln, so kann es höchstens 2^n Äquivalenzklassen von $\sim_\Gamma$ geben.

■

Satz 59 *Die modale Aussagenlogik* **K** *ist entscheidbar, d.h. es gibt einen Algorithmus, der für jede Formel $A \in Fml_{ALmod}$ entscheidet, ob A eine* **K***-Tautologie ist oder nicht.*

Beweis: Sei Γ die endliche Menge aller Teilformeln von $\neg A$. Wenn $\neg A$ überhaupt erfüllbar ist, dann nach Lemma 58 auch in einer Kripke Struktur mit höchstens 2^n Welten, wobei n die Kardinalität von Γ ist. Alle Kripke Stukturen mit höchstens 2^n Welten lassen sich endlich aufzählen, und es ist für jede dieser Strukturen entscheidbar, ob $\neg A$ in ihr wahr ist oder nicht.

■

Korollar 60
Jede modale Logik, die aus **K** *durch Hinzunahme von einem oder mehreren der Axiome*

$$\begin{array}{l} \Box p \to p \\ p \to \Box\Diamond p \\ \Box p \to \Box\Box p \end{array}$$

entsteht, besitzt die endliche Modelleigenschaft und ist entscheidbar.

Beweis: Man muß zeigen, daß der filtrierte Rahmen (G_Γ, R_Γ) reflexiv, symmetrisch bzw. transitiv ist, wenn der Ausgangsrahmen (G, R) diese Eigenschaft hatte. Für Reflexivität und Symmetrie ist das einfach nachzurechnen. Für die Transitivität klappt das nicht so einfach. Die Relation R_Γ auf der Quotientenstruktur ist im allgemeinen nicht transitiv. Wir betrachten statt $(G_\Gamma, R_\Gamma, v_\Gamma)$ die Kripkestruktur $(G_\Gamma, R^*_\Gamma, v_\Gamma)$, wobei R^*_Γ der transitive Abschluß von R_Γ ist. Als einzig interessanter Fall im induktiven Beweis des

Analogons von Lemma 58 (1) bleibt für jede Welt $w \in G$ und jede Formel der Form $\Box F$ zu zeigen

$$(G_\Gamma, R_\Gamma^*, v_\Gamma, [w]) \models \Box F \text{ gdw } (G, R, v, w) \models \Box F$$

Wir setzen zur Abkürzung $\mathcal{K}^* = (G_\Gamma, R_\Gamma^*, v_\Gamma)$ und $\mathcal{K} = (G, R, v)$.

Gelte zunächst $(\mathcal{K}^*, [w]) \models \Box F$. Um $(\mathcal{K}, w) \models \Box F$ nachzuweisen, betrachten wir ein $w' \in G$ mit wRw' in der Absicht $(\mathcal{K}, w') \models F$ zu zeigen. Aus wRw' folgt unmittelbar aus der Definition $wR_\Gamma w'$ und damit auch $wR_\Gamma^* w'$. Aus der Voraussetzung folgt jetzt $(\mathcal{K}^*, [w']) \models F$. Die Induktionsvoraussetzung liefert jetzt wie gewünscht $(\mathcal{K}, w') \models F$. Das war einfach. Die umgekehrte Implikation ist natürlich die kritische.

Hier setzen wir $(\mathcal{K}, w) \models \Box F$ voraus und wollen $(\mathcal{K}^*, [w]) \models \Box F$ beweisen. Dazu betrachten wir ein w' mit $[w]R_\Gamma^*[w']$. Denken wir daran daß diese Relation der transitive Abschluß von R_Γ ist und setzen wir auch schon die Definition von R_Γ ein, so erhalten wir die Existenz von Welten $u_0, \ldots u_k$ mit $[w] = [u_0]$ und $[w'] = [u_k]$ und die Existenz von Welten $g_i \in [u_i]$, $h_i \in [u_{i+1}]$ für $0 \leq i < k$ mit $g_i R h_i$. Man beachte, daß $[g_0] = [w]$, $[h_{k-1}] = [w']$ und $[g_{i+1}] = [h_i]$ gilt. Aus $(\mathcal{K}, w) \models \Box F$ folgt somit auch $(\mathcal{K}, g_0) \models \Box F$ und wegen der Transitivität von (G, R) auch $(\mathcal{K}, g_0) \models \Box\Box F$ Aus $g_0 R h_0$ folgt $(\mathcal{K}, h_0) \models \Box F$. Da, wie schon gesagt, h_0 und g_1 äquivalent sind erhalten wir auch $(\mathcal{K}, g_1) \models \Box F$. Wegen Transitivität folgt wieder $(\mathcal{K}, g_1) \models \Box\Box F$ und so weiter, bis wir schließlich $(\mathcal{K}, g_{k-1}) \models \Box F$ erreichen. Hieraus folgt wegen $g_{k-1} R h_{k-1}$ $(\mathcal{K}, h_{k-1}) \models F$ und wegen $[h_{k-1}] = [w']$ auch $(\mathcal{K}, w') \models F$. Nach Induktionsvoraussetzung erhalten wir daraus, wie gewünscht $(\mathcal{K}_\Gamma^*, [w']) \models F$.

■

Einen interessanten Überblick und tiefe Analyse von Entscheidbarkeitsresultaten in modalen Logiken findet man in [Grädel, 1999]

3.4.1 Übungsaufgaben

Übungsaufgabe 3.4.1 *Die in Definition 57 gegebene Definition für R_Γ ist nicht die einzig mögliche. Zeigen Sie, daß auch mit der Relation R^Γ Lemma 58 richtig bleibt:*

$$[g]R^\Gamma[h] \text{ gdw für alle } \Box B, B \in \Gamma \text{ folgt aus } (\mathcal{K}, g) \models \Box B \text{ auch } (\mathcal{K}, h) \models B$$

Übungsaufgabe 3.4.2 *Aus Satz 59 und Lemma 35 auf Seite 138 folgt, daß auch für* $A, B \in Fml_{ALmod}$ *die lokale Konsequenzrelation* $A \vdash_L^K B$ *entscheidbar ist. Zeigen Sie, daß auch die globale Konsequenzrelation für die modale Logik K entscheidbar ist.*

Übungsaufgabe 3.4.3 *Geben Sie eine Formelmenge* Γ *und eine transitive Kripke Struktur* $\mathcal{K} = (G, R, v)$ *an, so daß die Filtration* $\mathcal{K}_\Gamma$ *nicht transitiv ist.*

Übungsaufgabe 3.4.4 *Zeigen Sie, daß die modale Logik* $\mathbf{K} + \Box\Box p \to p$ *entscheidbar ist.*

3.5 Beschreibungslogiken

Beschreibungslogiken stammen aus dem Bereich der Wissensrepräsentationssysteme der künstlichen Intelligenz. Etwas präziser kann man ihren Ursprung zurückverfolgen zu den strukturierten Vererbungsnetzen von Brachman [Brachmann, 1977] [Brachmann, 1978], die ihre erste Realisierung in dem System KL-ONE [Brachmann & Schmolze, 1985] fanden. Im Bereich der künstlichen Intelligenz begegnete man in den achtziger Jahren der Wissensrepräsentation in einer logik-basierten Sprache, worunter man sich in der Regel die Prädikatenlogik erster Stufe vorstellte, mit Skepsis und auch einigen Mißverständnissen. Heute wird akzeptiert, daß Beschreibungssprachen als notationelle Varianten eines Teils der Prädikatenlogik aufgefasst werde können. Das Potential möglicher Anwendungen wird deutlich in dem Beitrag [Franconi *et al.*, 2000]. Eine umfassende Übersicht findet man in dem Handbuch [Baader *et al.*, 2003].

Es gibt unzählige Varianten von Beschreibungssprachen, wir konzentrieren uns hier auf die Sprache $\mathcal{ALC}$.

3.5.1 $\mathcal{ALC}$

Die Beschreibungslogik $\mathcal{ALC}$ kennt zwei elementare Grundbegriffe *Konzept* unf *Rolle* Eine Instanz der Beschreibungslogik $\mathcal{ALC}$ wird bestimmt durch die Festlegung eines endlichen Vorrats $\mathbf{C}$ von Symbolen für Konzepte und eines endlichen Vorrats $\mathbf{R}$ von Symbolen für Rollen. Die Vereinigung $\mathbf{V} = \mathbf{C} \cup \mathbf{R}$ nennen wir das Vokabular.

Definition 58 ($\mathcal{ALC}$-Ausdrücke)
Die Menge der $\mathcal{ALC}$-Ausdrücke über einem vorgegebenen Vokabular $\mathbf{V} = \mathbf{C} \cup \mathbf{R}$ ist induktiv definiert durch:

1. *jedes Konzeptsymbol C aus $\mathbf{C}$ ist ein $\mathcal{ALC}$-Ausdruck,*
2. *sind $C_1, \ldots, C_k$ $\mathcal{ALC}$-Ausdrücke, dann sind auch $C_1 \sqcap \ldots \sqcap C_k$, $C_1 \sqcup \ldots \sqcup C_k$ und $\neg C_1$ $\mathcal{ALC}$-Ausdrücke,*
3. *ist C ein $\mathcal{ALC}$-Ausdruck, R ein Rollensymbol aus $\mathbf{R}$, dann sind auch $\exists R.C$ und $\forall R.C$ $\mathcal{ALC}$-Ausdrücke.*

Manchmal nennt man die in der vorangegangen Definition 58 eingeführten Ausdrücke auch *Konzeptausdrücke*. Da in $\mathcal{A}LC$ keine anderen Ausdrücke auftreten verzichten wir auf diese Präzisierung.

Zur Festlegung der Bedeutung von $\mathcal{A}LC$-Ausdrücken bedarf es der Angabe einer Interpretation.

Definition 59 (Interpretation für $\mathcal{A}LC$)
Eine Interpretation für $\mathcal{A}LC$ über einem Vokabular $\mathbf{V}$ besteht aus einem Grundbereich von Objekten, der üblicherweise mit Δ bezeichnet wird, und dem Universum prädikatenlogischer Strukturen entspricht und einer Funktion $\mathcal{I}$, für die gilt:

1. *für jedes Konzeptsymbol $C \in \mathbf{C}$ ist $C^{\mathcal{I}}$ eine Teilmenge Δ*
2. *für jedes Rollensymbol $R \in \mathbf{R}$ ist $R^{\mathcal{I}}$ eine binäre Relation aus Δ, i.e. $R^{\mathcal{I}} \subseteq \Delta \times \Delta$.*

Ist eine Interpretation $\mathcal{I}$ gegeben, so ordnet die folgende Definition jedem $\mathcal{A}LC$-Ausdruck F eine Teilmenge $F^{\mathcal{I}}$ von Δ zu.

Definition 60 (Semantik von $\mathcal{ALC}$)
Sei $\mathcal{I}$ eine Interpretation.

1. $(C_1 \sqcap \ldots \sqcap C_k)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap \ldots \cap C_k^{\mathcal{I}}$
2. $(C_1 \sqcup \ldots \sqcup C_k)^{\mathcal{I}} = C_1^{\mathcal{I}} \cup \ldots \cup C_k^{\mathcal{I}}$
3. $(\neg C)^{\mathcal{I}} = \Delta \setminus C^{\mathcal{I}}$
4. $(\exists R.C)^{\mathcal{I}} = \{d \in \Delta \mid \text{ es gibt } (d,e) \in R^{\mathcal{I}} \text{ mit } e \in C^{\mathcal{I}}\}$,
5. $(\forall R.C)^{\mathcal{I}} = \{d \in \Delta \mid \text{ für alle } (d,e) \in R^{\mathcal{I}} \text{ gilt } e \in C^{\mathcal{I}}\}$

Beispiel 13
Sind M,F P Namen für Konzepte, die wir intuitiv als die Klasse aller Männer, aller Frauen bzw. aller Personen deuten und ist $R = ist\_direkter_V orfahre\_von$ eine Rolle, dann sind

$Va = M \sqcap \exists R.P \qquad Mu = F \sqcap \exists R.P$
$E = Va \sqcup Mu$

korrekt geformte Ausdrücke in $\mathcal{ALC}$, welche die Konzepte Vater, Mutter und Eltern einführen.

Übersetzt in prädikatenlogische Notation sehen diese Konzepte so aus:

$Va(x) = M(X) \wedge \exists y(R(x,y) \wedge P(y))$
$Mu(x) = F(x) \wedge \exists y(R(x,y) \wedge P(y))$
$E(x) = Va(x) \wedge Mu(x)$

Sind C_1, C_2 Ausdrücke einer Beschreibungslogik, so ist man daran interessiert festzustellen ob für alle Interpretationen $\mathcal{I}$

$C_1^{\mathcal{I}}$ dasselbe Konzept wie $C_2^{\mathcal{I}}$ ist	$C_1 = C_2$
$C_1^{\mathcal{I}}$ ein Teilmenge von $C_2^{\mathcal{I}}$ ist	$C_1 \sqsubseteq C_2$
$C_1^{\mathcal{I}}$ leer ist	$C_1 = \emptyset$

Wegen

$$C_1 = C_2 \text{ gdw } C_1 \sqsubseteq C_2 \wedge C_2 \sqsubseteq C_1$$

und

$$C_1 \sqsubseteq C_2 \text{ gdw } C_1 \sqcap \neg C_2 = \emptyset$$

genügt es sich auf die letzte Fragestellung $C = \emptyset$? zu beschränken. Die Beantwortung dieser Frage wird üblicherweise **Subsumtionsproblem** genannt. Die wichtige Beobachtung ist nun, daß das Subsumtionsproblem für $\mathcal{ALC}$ entscheidbar ist, ein Resultat was bei einer direkten Übersetzung in die Prädikatenlogik nicht sofort zu sehen ist. Eine Reihe von Algorithmen zur Beantwortung dieser Frage sind erdacht und implementiert worden. Wir werden hier eine Möglichkeit vorstellen, die Entscheidbarkeit des Subsumtionsproblems durch Reduktion auf das Erfüllbarkeitsproblem in einer multimodalen Logik zu beweisen.

Die Tatsache, da sich die Beschreibungslogik $\mathcal{A}LC$ auffassen lt als eine modale Logik wurde zuerst von Klaus Schild bemerkt in [Schild, 1991, Schild, 1993]. Wir übersetzen jeden Ausdruck F der Beschreibungslogik $\mathcal{A}LC$ in eine Formel F^* einer multi-modalen Logik, die für jede Rolle R modale Operatoren $\Box_R$ und $\Diamond_R$ enthält. Bei der Übersetzung wird jedes Konzeptsymbol C durch eine aussagenlogische Variable ersetzt, die wir der Einfachheit halber wieder mit C bezeichnen.

Definition 61
Jedem $\mathcal{ALC}$-Ausdruck F wird die multi-modale Formel F^* wie folgt zugeordnet:

$$
\begin{array}{lcll}
F^* & = & F & \textit{falls } F \textit{ ein Konzeptsymbol ist} \\
(F_1 \sqcap F_2)^* & = & (F_1^* \wedge F_2^*) & \\
(\neg F)^* & = & \neg F^* & \\
(\forall R.F)^* & = & \Box_R F^* & \\
(\exists R.F)^* & = & \Diamond_R F^* &
\end{array}
$$

Als nächstes bemerken wir, daß eine Interpretation $\mathcal{I}$ für $\mathcal{ALC}$ in eindeutiger Weise einer multi-modalen Kripke Stuktur $\mathcal{K}^\mathcal{I}$ entspricht. Die Menge der möglichen Welten von $\mathcal{K}^\mathcal{I}$ entspricht dabei dem Universum Δ von $\mathcal{I}$. Jede Rolle R gibt Anlaß zu einer Zugänglichkeitsrelation $R^\mathcal{I}$ auf Δ, die für die Semantik von $\Box_R$ und $\Diamond_R$ zuständig ist. Für $d \in \Delta$ wird schließlich die Bewertung $v(d, C)$ auf *wahr* gesetzt genau dann, wenn $d \in C^\mathcal{I}$ zutrifft.

Lemma 61

1. *Für jeden $\mathcal{ALC}$-Ausdruck F und jede Interpretation $\mathcal{I}$ gilt*

$$F^\mathcal{I} = \{d \in \Delta \mid (\mathcal{K}^\mathcal{I}, g) \models F^*\}$$

2. *Es gibt eine Interpretation $\mathcal{I}$ und ein $d \in \Delta$ mit $d \in F^\mathcal{I}$ genau dann, wenn F^* erfüllbar ist.*

3. *Es gilt $F_1 \sqsubseteq F_2$ genau, dann wenn $(F_1 \sqcap \neg F_2)^*$ unerfüllbar ist.*

Beweis:
zu 1: Der Beweis wird über den Aufbau des Ausdrucks F geführt. Ist F ein Konzeptname, dann ergibt sich nach Definition von $\mathcal{K}^\mathcal{I}$

$$
\begin{array}{rcl}
\{d \in \Delta \mid (\mathcal{K}^\mathcal{I}, g) \models F^*\} & = & \{d \in \Delta \mid v(d, F) = \textit{wahr}\} \\
& = & F^\mathcal{I}
\end{array}
$$

Die Induktion über die Operatoren $\sqcap$,$\sqcup$ und $\neg$ ist einfach und wird hier weggelassen. Wir betrachten noch den Fall $F = \forall R.F_1$

$$\begin{array}{rcl}\{d \in \Delta \mid (\mathcal{K}^{\mathcal{I}}, g) \models (\forall R.F_1)^*\} & = & \{d \in \Delta \mid (\mathcal{K}^{\mathcal{I}}, g) \models \Box_R F_1^*\} \\ & = & \{d \in \Delta \mid \text{ für alle } e \text{ mit } R(d,e) \text{ gilt} \\ & & \quad (\mathcal{K}^{\mathcal{I}}, e) \models F_1^*\} \\ & = & \{d \in \Delta \mid \text{ für alle } e \text{ mit } R(d,e) \text{ gilt} \\ & & \quad e \in F_1^{\mathcal{I}}\} \\ & = & (\forall R.F_1)^{\mathcal{I}}\end{array}$$

zu 2: folgt unmittelbar aus Teil 1.
zu 3:
$F_1 \sqsubseteq F_2$ gilt genau dann, wenn für alle $\mathcal{I}$ und alle $d \in \Delta$ gilt $d \notin (F_1 \sqcap \neg F_2)^{\mathcal{I}}$. Nach Teil 1 ist das gleichbedeutend mit $(K^{\mathcal{I}}, d) \not\models F_1^* \wedge \neg F_2^*$ für alle $\mathcal{I}$ und alle $d \in \Delta$. Das besagt nichts anderes, als daß $(F_1 \sqcap \neg F_2)^*$ nicht erfüllbar ist.

■

3.5.2 $\mathcal{SHIQ}$

Für praktische Anwendungen, so stellte sich schnell heraus, ist die Ausdrucksstärke von $\mathcal{ALC}$ zu schwach. Unter den vielen Erweiterungen betrachten wir die Sprache $\mathcal{SHIQ}$

Die erfolgreichste Implementierung einer Beschreibungslogik zum gegenwärtigen Zeitpunkt ist das FaCT[1] System, beschrieben in [Patel-Schneider & Horrocks, 1999].

Ein Vokabular für $\mathcal{SHIQ}$ besteht wieder aus einer Menge $\mathbf{C}$ von Konzeptsymbolen und einer Menge $\mathbf{R}^0$ von atomaren Rollensymbolen. In $\mathbf{C}$ ist stets das unverselle Konzept $\top$ enthalten. Neu ist die explizite Angabe einer Teilmenge $\mathbf{R}^0_t \subseteq \mathbf{R}$. Von den atomaren Rollen in $\mathbf{R}^0_t$ wird später verlangt werden, daß sie transitiv sind. Die Menge aller Rollen $\mathbf{R}$ enthält alle atomaren Rollen und für jede atomare Rolle $R \in \mathbf{R}^0$ ihre inverse Rolle R^-.

Definition 62 (Rollenhierarchie)
Eine Rollenhierarchie $\mathcal{H}$ ist eine endliche Menge von Formeln der Art $R_1 \sqsubseteq R_2$ für Rollensymbole $R_i \in \mathbf{R}$.

[1] http://www.cs.man.ac.uk/ horrocks/FaCT/

Definition 63 ($\mathcal{SHIQ}$-Ausdrücke)
Die Menge der $\mathcal{SHIQ}$-Ausdrücke über einem vorgegebenen Vokabular $\mathbf{V} = \mathbf{C} \cup \mathbf{R}$ ist induktiv definiert durch:

1. *jedes Konzeptsymbol C aus $\mathbf{C}$ ist ein $\mathcal{SHIQ}$-Ausdruck,*
2. *das Symbol $\top$ ist ein $\mathcal{SHIQ}$-Ausdruck,*
3. *sind $C_1, \ldots, C_k$ $\mathcal{SHIQ}$-Ausdrücke, dann sind auch $C_1 \sqcap \ldots \sqcap C_k$, $C_1 \sqcup \ldots \sqcup C_k$ und $\neg C_1$ $\mathcal{SHIQ}$-Ausdrücke,*
4. *ist C ein $\mathcal{SHIQ}$-Ausdruck, R ein Rollensymbol aus $\mathbf{R}$, dann sind auch $\exists R.C$ und $\forall R.C$ $\mathcal{SHIQ}$-Ausdrücke,*
5. *ist R ein einfacher Rollenbezeichner aus $\mathbf{R}$, C ein $\mathcal{SHIQ}$-Ausdruck und n eine natürliche Zahl, dann sind $\leq nR.C$ und $\geq nR.C$ $\mathcal{SHIQ}$-Ausdrücke. Dabei heißt ein Rollenbezeichner R einfach, wenn R nicht transitive ist, d.h. wenn $R \notin I\!R^0_t$ gilt und überdies R keine transitive Unterrolle besitzt.*

Beispiel 14
Zusätzlich zu den Konzepten und Rollen aus Beispiel 13 benutzen wir noch das Konzept W, das intuitiv für alle weiblichen Personen stehen soll. Der Ausdruck

$$C = E \sqcap \ \geq 2R.W$$

bezeichnet die Menge aller Eltern, die mindestens zwei Töchter haben.

Wegen der Einschränkung in Klausel 5 hängt die Definition der syntaktisch erlaubten $\mathcal{SHIQ}$-Ausdrücke von einer vorher fixierten Rollenhierachie $\mathcal{H}$ ab. Wir verzichten aus Gründen eine besseren Lesbarkeit darauf, diese Abhängigkeit in der Notation sichtbar zu machen.

Definition 64 (Interpretation für $\mathcal{SHIQ}$)
Eine Interpretation für das Vokabular $\mathbf{V}$ besteht aus einem mit Δ bezeichneten Grundbereich von Objekten und einer Funktion $\mathcal{I}$ mit folgenden Eigenschaften:

1. *für jedes Konzeptsymbol $C \in \mathbf{C}$ ist $C^{\mathcal{I}}$ eine Teilmenge von Δ,*

2. *für das spezielle Symbol* $\top \in \mathbf{C}$ *gilt* $\top^{\mathcal{I}} = \Delta$,

3. *für jedes atomare Rollensymbol* $R \in \mathbf{R}^0$ *ist* $R^{\mathcal{I}}$ *eine binäre Relation auf* Δ, *i.e.* $R^{\mathcal{I}} \subseteq \Delta \times \Delta$,

4. *für jedes atomare Rollensymbol* $R \in \mathbf{R}^0_t$ *ist* $R^{\mathcal{I}}$ *sogar eine transitive Relation auf* Δ,

5. *für jedes atomare Rollensymbol* $R \in \mathbf{R}^0$ *ist* $(R^-)^{\mathcal{I}}$ *die inverse Relation zu* $R^{\mathcal{I}}$, *i.e.*
$$(d_1, d_2) \in (R^-)^{\mathcal{I}} \Leftrightarrow (d_2, d_1) \in R^{\mathcal{I}}$$

Definition 65 (Semantik von $\mathcal{SHIQ}$-Ausdrücken)
Sei $\mathcal{I}$ eine Interpretation. Die Bedeutung von $\mathcal{SHIQ}$-Ausdrücken, die nicht schon in Definition 60 erklärt wurde, ist wie folgt

1. $(\leq nR.C)^I = \{d \in \Delta \mid \#\{e \mid (d,e) \in R^{\mathcal{I}} \text{ und } e \in C^{\mathcal{I}}\} \leq n\}$

2. $(\geq nR.C)^I = \{d \in \Delta \mid \#\{e \mid (d,e) \in R^{\mathcal{I}} \text{ und } e \in C^{\mathcal{I}}\} \geq n\}$

Definition 66
Sei $\mathcal{H}$ eine Rollenhierarchie. Es wird im folgenden vorausgesetzt, daß alle auftretenden Interpretationen $\mathcal{I}$ die Rollenhierarchie respektieren, d.h. es gilt $R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$ für jede Formel $R_1 \sqsubseteq R_2 \in \mathcal{H}$.

1. *Eine endliche Menge $\mathcal{T}$ von Formeln der Form $C_1 \sqsubseteq C_2$ für $\mathcal{SHIQ}$-Ausdrücke C_1, C_2 heißt eine* Terminologie

2. *Eine Interpretation $\mathcal{I}$ heißt ein* Modell *einer Terminologie $\mathcal{T}$, wenn $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$ für alle $C_1 \sqsubseteq C_2 \in \mathcal{T}$ gilt.*

3. *Ein $\mathcal{SHIQ}$-Ausdruck C ist erfüllbar bezüglich $\mathcal{T}$, wenn es ein Modell $\mathcal{I}$ von $\mathcal{T}$ gibt mit $C^{\mathcal{I}} \neq \emptyset$*

4. *D subsumiert C bezüglich $\mathcal{T}$, wenn für jedes Modell $\mathcal{I}$ von $\mathcal{T}$ gilt mit $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.*

Satz 62
Das Erfüllbarkeits- und das Subsumptionsproblem für $\mathcal{SHIQ}$ ist entscheidbar.

Beweis siehe [Horrocks *et al.*, 2000].

Läßt man in Definition 63 , Klausel 5 die Einschränkung auf einfache Rollen fallen, so wird das Erfüllbarkeitsproblem unentscheidbar. Das gilt sogar, dann wenn man nur Kardinalitätsformeln der Form $\leq nR.\top$ und $\geq nR.\top$ zuläßt, siehe ebenfalls [Horrocks *et al.*, 2000].

3.5.3 Übungsaufgaben

Übungsaufgabe 3.5.1 *fehlt noch*

3.6 Modale Quantorenlogiken

Im nächsten Kapitel wollen wir Tableaukalkül für die modale Quantorenlogik $\mathcal{K}$ vorstellen, einschließlich der zugehörigen Korrektheits- und Vollständigkeitsbeweise. Wir geben am Ende einige Hinweise, wie man durch leichte Modifikationen Tableaukalküle für andere modale Logiken erhalten kann. Weitere Hinweise findet man in dem Buch [Fitting, 1983], dem wir in der Ausarbeitung dieses Kapitels weitgehend folgen. Eine neuere, erweiterte Darstellung von Tableaukalkülen für modale Logiken findet man in [Fitting, 1991]. Zunächst jedoch ist die Einführung von Quantoren in einer modalen Logik zu erklären.

3.6.1 Modale Prädikatenlogiken

In Abschnitt 3.1 haben wir Syntax und Semantik modaler Aussagenlogiken eingeführt. Wir wollen jetzt die Erweiterung auf modale Quantorenlogiken vorstellen. Die Syntax bereitet keinerlei Probleme. In den folgenden Definitionen ist stillschweigend ein fest vorgegebenes Vokabular V vorausgesetzt.

Definition 67 *Die Menge , Fml_{Qmod}, aller Formeln der modalen Prädikatenlogik ist die kleinste Menge von Formeln, die alle Klauseln der Definition 34 erfüllt und zusätzlich*

4. *Ist $F \in Fml_{Qmod}$ und x eine Individuenvariable, dann sind auch $\forall xF$ und $\exists xF$ Formeln in Fml_{Qmod}.*

Die Semantik für die modale Prädikatenlogik wird durch eine Erweiterung der aussagenlogischen Kripke Strukturen gegeben. Die neuen Strukturen wollen wir einfach Kripke Strukturen nennen, ohne Zusatz.

Definition 68 (Kripke Strukturen)
Eine **Kripke Struktur** $\mathcal{K} = (G, R, M)$ *besteht aus einem Kripke Rahmen (G, R) und einer Abbildung M, die jeder möglichen Welt $g \in G$ eine gewöhnliche prädikatenlogische Struktur $M(g)$ zuordnet, so daß für alle g_1, g_2 in G mit $R(g_1, g_2)$ gilt*

- *das Universum U_1 von $M(g_1)$ eine Teilmenge des Universums U_2 von $M(g_2)$ und*

- *auf der kleineren Menge U_1 stimmen die Interpretationen der Funktionszeichen in $M(g_1)$ und $M(g_2)$ überein.*

Die in dieser Definition genannte Bedingung an $M(g_1)$ und $M(g_2)$ wird üblicherweise durch die Redeweise „$M(g_1)$ ist eine **Unteralgebra** von $M(g_2)$“beschrieben. Man beachte,daß von den Interpretationen der Relationszeichen in $M(g_1)$ und $M(g_2)$ keine Einschränkung verlangt wird. Wir werden uns hauptsächlich mit dem folgenden einfachen Spezialfall von Kripke Stukturen beschäftigen:

Definition 69
Eine Kripke Struktur $\mathcal{K} = (G, R, M)$ heißt eine Kripke Struktur **mit konstantem Universum**, *wenn für alle $g \in G$ die Struktur $M(g) = \mathcal{M}$ dasselbe, konstante Universum M hat.*

Es gibt eine Unzahl von Varianten für die Definition von Kripke Strukturen für die modale Prädikatenlogik. In einer Variante wird z.B. auf die Unterstrukturbedingung verzichtet, oder es wird zwar eine Teilmengenbedingung für die Universa aber keine Verträglichkeit der Interpretationen der Funktions- und Relationszeichen gefordert oder sogar die Abgeschlossenheit der Universa unter den Funktionen des Vokabulars V fallengelassen. Einen umfangreichen Überblick dazu findet man in [Garson, 1984].

Die Modellbeziehung zwischen einer Kripke Struktur und einer Formel ohne freie Variablen kann jetzt aus der Definition 38 erhalten werden, in dem die Klausel für aussagenlogische Variable ersetzt wird durch eine Klausel über atomare Formeln und die gewohnte Klausel für die Interpretation der Quantoren hinzugefügt wird.

Definition 70
Sei $\mathcal{K} = (G, R, M)$ eine Kripke Struktur und F eine Formel in Fml_{Qmod}, dann wird die Relation $(\mathcal{K}, g) \models F$, gelesen als „ F ist wahr (oder gültig) in derWelt g der Kripke Struktur $\mathcal{K}$“, wie folgt definiert.
Zur Vereinfachung der Notation schreiben wir $g \models F$ anstelle von $(\mathcal{K}, g) \models F$ und bezeichnen das Universum der Struktur $M(g)$ mit U_g.

1	$g \models r(t_1, \dots t_k)$	*gdw*	$M(g) \models r(t_1, \dots t_k)$ *für atomare Formeln* $r(t_1, \dots t_k)$
2	$g \models F \wedge H$	*gdw*	$g \models F$ *und* $g \models H$
3	$g \models F \vee H$	*gdw*	$g \models F$ *oder* $g \models H$
4	$g \models F \rightarrow H$	*gdw*	(*nicht* $g \models F$) *oder* $g \models H$
5	$g \models \neg F$	*gdw*	*nicht* $g \models F$
6	$g \models \Box F$	*gdw*	*für alle* h *mit* $R(g, h)$ *gilt* $h \models F$
7	$g \models \Diamond F$	*gdw*	*es gibt ein* h *mit* $R(g, h)$ *so daß* $h \models F$
8	$g \models \forall x F(x)$	*gdw*	*für alle* $d \in U_g$ *gilt* $g \models F(d)$
9	$g \models \exists x F(x)$	*gdw*	*es gibt ein* $d \in U_g$ *mit* $g \models F(d)$

3.6.2 Reduktion auf Prädikatenlogik erster Stufe

In diesem Abschnitt soll gezeigt werden, wie sich die modale Logik auf die Logik des Prädikatenkalküls erster Stufe reduzieren läßt. Wir beschränken uns dabei auf modale Logiken mit konstantem Universum. Einem modallogischen Vokabular V wird dabei ein prädikatenlogisches Vokabular V_{Red} zugeordnet, jeder Formel $A \in \; Mfml(V)$ wird in eine Formel $\mathrm{Red}(A)$ des PK1 im Vokabular V_{Red} transformiert und zu jeder modallogischen Struktur $K = < G, R, M >$ wird eine V_{Red}-Struktur $\mathrm{Red}(K)$ konstruiert, so daß gilt:

$$(K, g) \models A(a_1, \ldots, a_k) \text{ gdw } \mathrm{Red}(K) \models \mathrm{Red}(A)(a_1, \ldots, a_k, g)$$

Definition 71 ($V_{\mathbf{Red}}$)

- *Für jedes n-stellige Relationszeichen $p(x_1, \ldots, x_n)$ in V enthält V_{Red} ein (n+1)-stelliges Relationszeichen $p_{Red}(x_1, \ldots, x_n, x_{n+1})$.*
- *V_{Red} enthält eine neues binäres Relationszeichen $r(x, y)$ und ein neues einstelliges Relationszeichen $w(x)$.*
- *Die Funktions- und Konstantenzeichen von V_{Red} sind dieselben, wie für V.*

Definition 72 (Red(A))

1. *Red($p(t_1, \ldots, t_n)$) = $p_{Red}(t_1, \ldots, t_n, y) \wedge w(y)$*
2. *Red($A \wedge B$) = Red(A) $\wedge$ Red(B)*
3. *Red($\neg A$) = $\neg$Red(A)*
4. *Red($\exists xA$) = $\exists x(\neg w(x) \wedge$ Red(A)) item Red($\forall xA$) = $\forall x(\neg w(x) \rightarrow$ Red(A))*
5. *Red($\Box A$) = $\forall x(r(y, x) \rightarrow \mathrm{Red}(A)(\frac{x}{y})$, wobei x eine Variable ist, die nicht frei in Red(A) vorkommt.*
6. *Red($\Diamond A$) = $\exists x(r(y, x) \wedge \mathrm{Red}(A)(\frac{x}{y})$, wobei x eine Variable ist, die nicht frei in Red(A) vorkommt.*

Die Formel Red(A) enthält stets eine freie Variable mehr als A. Diese zusätzliche freie Variable wird in allen Formeln mit y bezeichnet. Dabei wird vorausgesetzt, daß y in keiner Formel A benutzt wird. Ebenso wird stillschweigend vorausgesetzt, daß die neue Variable x, die in Red($\Box A$) und Red($\Diamond A$) benutzt wird, neu ist.

Beispiel 15 () *Red(*$\forall x \Box p(x) \rightarrow \Box \forall x p(x)$*):*

$$\begin{array}{l} \forall x(\neg w(x) \rightarrow [\forall z(r(y,z) \rightarrow (p(x,z) \wedge w(z))]) \\ \rightarrow \\ \forall z(r(y,z) \rightarrow \forall x(\neg w(x) \rightarrow (p(x,z) \wedge w(z))) \end{array}$$

Definition 73 (Red(K))
Sei $K = (G, R, M)$ eine Kripke Struktur im Vokabular V mit Universum $U = U_g$ für alle $g \in G$.

Das Universum von Red(K) sei $G \cup U$.

Sei g_0 ein beliebiges Element aus G. Für ein Funktionszeichen f aus V setzen wir:

$$f^{Red}(a_1, \ldots, a_k) = \begin{cases} f^{M(g)}(a_1, \ldots, a_k) & \textit{falls alle } a_i \in U \\ g_0 & \textit{sonst} \end{cases}$$

Für ein Relationszeichen p aus V setzen wir:

Red(K)$\models p_{Red}(a_1, \ldots, a_n, b)$ gdw $b \in G, a_1, \ldots, a_n \in U$ und $(K, b) \models p(a_1, \ldots, a_n)$.

Für die neuen Relationszeichen r und w soll gelten:

$$\begin{array}{lll} \textit{Red(K)} \models r(b,c) & \textit{gdw} & b, c \in G \textit{ und } bRc. \\ \textit{Red(K)} \models w(b) & \textit{gdw} & b \in G. \end{array}$$

Lemma 63 (Reduktionslemma)
Sei $A(a_1, \ldots, a_k) \in Mfml$, $K =< G, R, M >$ eine Kripke Struktur, $g \in G$ und $a_1, \ldots, a_k \in M(g)$, dann gilt:

$$(K, g) \models A(a_1, \ldots, a_k) \textit{ gdw } \textit{Red}(K) \models \textit{Red}(A)(a_1, \ldots, a_k, g)$$

Beweis: Induktion über die Komplexität der Formeln A.
Ist $A(x_1, \ldots, x_k) = p(x_1, \ldots, x_k)$ so folgt die Behauptung des Lemmas direkt aus der Definition der Interpretation des neuen Relationszeichens $p_{\text{Red}}(x_1, \ldots, x_k, y)$ in Red(K).
Im Fall $A(x_1, \ldots, x_k) = \exists x_0 A_0(x_0, x_1, \ldots, x_k)$ argumentieren wir zunächst für die Implikation von links nach rechts.
Aus $(K, g) \models A(a_1, \ldots, a_k)$ folgt die Existenz eines Elements $a_0 \in U$ mit $(K, g) \models A_0(a_0, a_1, \ldots, a_k)$, woraus nach der Induktionsvoraussetzung Red(K)$\models$ Red($A_0(a_0, a_1, \ldots, a_k, g)$) folgt. Nach der Interpretation von w in Red(K) gilt auch Red(K)$\models \neg w(a_0)$ und damit auch Red(K)$\models \exists x_0(\neg w(x_0) \wedge A_0(x_0, a_1, \ldots, a_n, g))$, i.e. Red($K$)$\models$ Red(A).
Zum Beweis der umgekehrten Implikation gehen wir aus von der Annahme

$$\text{Red}(K) \models \text{Red}A(a_1, \ldots, a_n, g),$$

d.h. von

$$\text{Red}(K) \models \exists x_0(\neg w(x_0) \wedge \text{Red}(A_0(x_0, a_1, \ldots, a_n, g))).$$

Es gibt also ein $a_0 \in U$ mit Red(K)$\models$ Red$A_0(a_0, a_1, \ldots, a_n, g)$. Nach Induktionsvoraussetzung folgt

$$(K, g) \models A_0(a_0, a_1, \ldots, a_k)$$

und somit auch $(K, g) \models \exists x_0 A_0(x_0, a_1, \ldots, a_k)$.

■

Satz 64 (Reduktionssatz)
Seien $A \in Mfml$, $\Sigma \subset Mfml$ ohne freie Variable und Red(Σ) $=$ $\{\forall y(w(y) \rightarrow \mathit{Red}(B)) \ : \ B \in \Sigma\}$

1. *Ist $\forall y(w(y) \rightarrow \mathit{Red}(A))$ eine prädikatenlogische Tautologie, dann ist A eine* **K***-Tautologie.*

2. *Gilt Red(Σ) $\vdash \forall y(w(y) \rightarrow \mathit{Red}(A)$, dann gilt auch $\Sigma \vdash_G^{\mathbf{K}} A$.*

Beweis:
zu 1 Sei $K =< G, R, M >$ eine Kripke-Struktur. Wir müssen für jedes $g \in G$ zeigen $(K, g) \models A$. Nach Voraussetzung gilt Red(K) $\models \forall y(w(y) \rightarrow$

Red(A), also auch Red(K) $\models$ Red(A)(g). Mit Lemma 63 folgt die gewünschte Aussage.

zu 2 Sei $K =< G, R, M >$ wieder eine Kripke-Struktur. Wir nehmen an, daß für alle $g \in G$ gilt $(K, g) \models \Sigma$ und wollen zeigen, daß A in K gültig ist. Wegen Lemma 63 folgt für jede Formeln $B \in \Sigma$ aus dieser Voraussetzung zunächst Red(K) $\models$ Red(B)(g) für alle $g \in G$, und damit auch Red(K) $\models$ $\forall y(w(y) \rightarrow \mathrm{Red}(B)(y))$. Nach der Prämisse der zu beweisenden Behauptung folgt daraus Red(K) $\models \forall y(w(y) \rightarrow \mathrm{Red}(A)(y))$, d.h. Red($K$) $\models$ Red(A)(g)) für alle $g \in G$. Nach Lemma 63 folgt hieraus die Gültigkeit von A in K.

■

Die umgekehrten Implikationen in Satz 64 sind nicht richtig. Das liegt daran, daß nicht jede V_{Red}-Struktur $\mathcal{M}$ in der Form Red(K) = $\mathcal{M}$ dargestellt werden kann. Dieser Mangel läßt sich beheben.

Definition 74
Mit Σ_{Kripke} bezeichnen wir die folgende Menge prädikatenlogischer Formeln im Vokabular V_{Red}:

$$\exists x w(x)$$

$$\forall x \forall y (r(x, y) \rightarrow (w(x) \wedge w(y)))$$

Für jedes Relationszeichen $p \in V$:

$$\forall x_1 \dots x_k, y(p_{Red}(x_1 \dots x_k, y) \rightarrow \neg w(x_1) \wedge \dots \wedge \neg w(x_k) \wedge w(y))$$

Für jedes Funktionszeichen $f \in V$:

$$\forall x_1 \dots x_k(\neg w(x_1) \wedge \dots \wedge \neg w(x_k) \rightarrow \neg w(f(x_1 \dots x_k)))$$
$$\forall x_1 \dots x_k(w(x_1) \vee \dots \vee w(x_k) \rightarrow f(x_1 \dots x_k) = c) \wedge w(c)$$

Hierbei ist c ein neues Konstantenzeichen.

Lemma 65 *Sei $\mathcal{M}$ eine V_{Red}-Struktur. Dann gibt es eine Kripke Struktur K im Vokabular V mit Red(K) = $\mathcal{M}$*

gdw

$\mathcal{M} \models \Sigma_{Kripke}$

Satz 66 (Zweiter Reduktionssatz)
Seien $A \in Mfml$, $\Sigma \subset Mfml$ ohne freie Variable und $Red(\Sigma) = \{\forall y(w(y) \rightarrow Red(B)) \ : \ B \in \Sigma\}$

1. *A ist eine* **K**-*Tautologie*

   gdw

   $\Sigma_{Kripke} \vdash_{PK} \forall y(w(y) \rightarrow Red(A))$

2. $Red(\Sigma) \cup \Sigma_{Kripke} \vdash_{PK} \forall y(w(y) \rightarrow Red(A))$

   gdw

   $\Sigma \vdash_G^{\mathbf{K}} A.$

Beweis: Analog zum Beweis von Satz 64 mit Hilfe von Lemma 65.

■

3.6.3 Übungsaufgaben

Übungsaufgabe 3.6.1 *Zeigen Sie, daß für jede Welt g einer Kripke Struktur mit konstantem Universum für jede Formel $F(x)$ mit der freien Variablen x gilt*

$$g \models \forall x \Box F(x) \rightarrow \Box \forall x F(x)$$

Diese Formel ist in der Literatur unter dem Namen Barcansche Formel bekannt.

Übungsaufgabe 3.6.2 *Geben Sie eine Formel A der modalen Quantorenlogik an, so daß jedes Modell $\mathcal{K} = (G, R, M)$ von A eine unendliche Menge G von Welten besitzt.*

Übungsaufgabe 3.6.3 *Gibt es eine modallogische Formel A, so daß A in allen Kripke Strukturen (G, R, M) gilt, für die R der Einschränkung*

$$\forall xyz(xRz \wedge yRz \rightarrow x = y)$$

genügt, und A gilt aber nicht in allen Kripke trukturen?

Übungsaufgabe 3.6.4 *Diese Aufgabe dient der genaueren Untersuchung von Definition 72, insbesondere der Frage, wozu die Relation $w(x)$ gebraucht wird. Stellen Sie sich vor die Transformation von A in Red(A) wäre wie in Definition 72, jedoch ohne $w(x)$ zu benutzen. Diese Transformation wollen wir für die Zwecke dieser Übungsaufgabe mit Red′ bezeichnen. Insbesondere gilt also*

$$Red'(\exists x A_0) = \exists x Red'(A_0)$$

Finden Sie eine Formel und eine Kripke Struktur $K =< G, R, M >$, so daß Lemma 63 falsch wird.

Übungsaufgabe 3.6.5 *Zeigen Sie, daß die umgekehrte Implikation in Satz 64 nicht richtig ist. Finden Sie also eine Formel F, so daß Red(F) eine Tautologie ist, aber F keine modallogische Tautologie ist.*

Übungsaufgabe 3.6.6 *Berechnen Sie Red(A) für die folgenden Formeln der modalen Aussagenlogik. Machen Sie sich vorher klar, welche Konsequenz die Beschränkung auf die Aussagenlogik für den Übersetzungsprozeß insgesamt hat.*

1. $\Box\Box p \rightarrow \Box p$
2. $\Box\Diamond p \rightarrow \Diamond\Box p$
3. $p \rightarrow \Box p$

Übungsaufgabe 3.6.7 *Formulieren und beweisen Sie die Variante von Satz 66 Teil 2 für die* **lokale** *Konsequenzbeziehung.*

Übungsaufgabe 3.6.8 *In [Mortimer, 1975] wurde bewiesen, daß das Fragment FO^2 der Prädikatenlogik, in dem nur zwei Variablensymbole erlaubt sind, entscheidbar ist. (Genauer wurde gezeigt, daß FO^2 die endliche Modelleigenschaft hat.) Eine Übersicht über neuere und verwandte Resultate findet man in [Grädel & Otto, 1999].*

(a) *Zeigen Sie, daß die Reduktion Red aus Definition 72 so gewählt werden kann, daß $Red_1(A)$ in FO^2 liegt. Hinweis: Im aussagenlogischen Fall wird das Hilfsprädikat $w(.)$ nicht benötigt.*

Mit (a) erhalten wir einen zweiten Beweis für die Entscheidbarkeit der modale Aussagenlogik **K**, *59. Allerdings hat diese Technik ihre Grenzen. Schon die Entscheidbarkeit der modalen Aussagenlogik mit transitiver Zugänglichkeitsrelation R kann auf diese Weise nicht bewiesen werden, da die Transitivität nicht in dem Fragment FO^2 ausgedrückt werden kann.*

(b) *Zeigen Sie, daß es die Transitivität einer Relation $r(x, y)$ in FO^2 nicht ausgedrückt werden kann.*
Hinweis: Versuchen Sie aus der Annahme, Transitivität sei in FO^2 ausdrückbar, einen Widerspruch zur endlichen Modelleigenschaft von FO^2 herzuleiten.

3.7 Kalküle

3.7.1 Tableaukalküle

In einem vorangegangenen Kapitel haben wir schon einen Tableaukalkül für die dreiwertige Logik $\mathcal{L}_3$kennengelernt, so daß wir zum allgemeinen Rahmen solcher Kalküle nichts mehr sagen müssen.

Wir stellen einen Tableaukalkül mit signierten Formeln vor, die Vorzeichen sind, da wir in einer zwar modalen, aber doch zweiwertigen Logik arbeiten, F und T.

Es existieren recht unterschiedliche Versionen von Tableaukalkülen (siehe wieder [Fitting, 1983]). Wir haben uns hier für die Version mit Präfixen entschieden. Ein **Präfix** σ ist dabei eine endliche Folge natürlicher Zahlen. In den Tableaux werden also signierte Formeln mit Präfix, d.h. Gebilde der Form σZA auftreten, wobei σ ein Präfix, $Z \in \{F, T\}$ und A eine modallogische Formel ist. In diesem frühen Stadium der Erklärung läßt sich die Rolle der Präfixe erklären als Namen für verschiedene Welten einer Kripke-Struktur. Wir nennen ein Präfix σ_1 **zugänglich** von σ, wenn $\sigma_1 = \sigma n$ für ein geeignetes n ist. Diese Definition der Zugänglichkeit funktioniert für die Modallogik **K**, siehe Abbildung 3.1. Für andere Modallogiken ist die Zugänglichkeitsrelation geeignet zu variieren. Falls erforderlich reden wir genauer von **K**-Zugänglichkeit. Ein Problem bei der Erklärung und noch vielmehr bei der Darstellung der Korrektheits- und Vollständigkeitsbeweise für den dreiwertigen Tableaukalkül war die große Zahl der Tableauregeln. In dem darzustellenden modalen Tableaukalkül werden zwar weniger Regeln benötigt, aber immer noch mehr, als man in einem Blick überschauen kann. Um diesem didaktischen Problem zu begegnen, führen wir, wie in [Smullyan, 1968] und [Fitting, 1983], eine Klassifikation signierter Formeln ein in die Klassen der α-, β-, γ-, δ-, ν- und π-Formeln.

α-Formeln sind:	$TA \wedge B, FA \vee B, FA \rightarrow B, F\neg A$
β-Formeln sind:	$TA \vee B, FA \wedge B, TA \rightarrow B, T\neg A$
γ-Formeln sind:	$T\forall xA(x), F\exists xA(x)$
δ-Formeln sind:	$T\exists xA(x), F\forall xA(x)$
ν-Formeln sind:	$T\Box A, F\Diamond A$
π-Formeln sind:	$T\Diamond A, F\Box A$

indexFormeln!γ-

Jeder α- und β-Formel ordnen wir zwei Nachfolgerformeln α_1, α_2 bzw. β_1, β_2 zu:

α	α_1	α_2
$TA \wedge B$	TA	TB
$FA \vee B$	FA	FB
$FA \to B$	TA	FB
$F\neg A$	TA	TA

β	β_1	β_2
$TA \vee B$	TA	TB
$FA \wedge B$	FA	FB
$TA \to B$	TB	FA
$T\neg A$	FA	FA

Den γ-, δ-, ν-, π-Formeln wird jeweils eine Nachfolgerformel $\gamma_0, \ldots, \pi_0$ zugeordnet.

γ	$\gamma_0(x)$
$T\forall xA(x)$	$TA(x)$
$F\exists xA(x)$	$FA(x)$

δ	$\delta_0(x)$
$T\exists xA(x)$	$TA(x)$
$F\forall xA(x)$	$FA(x)$

ν	ν_0
$T\Box A$	TA
$F\Diamond A$	FA

π	π_0
$T\Diamond A$	TA
$F\Box A$	FA

Die angegebene Einteilung in α-, β-, γ-, δ-, ν- und π-Formeln ist offensichtlich erschöpfend, d.h. jede signierte, prädikatenlogische modale Formel gehört genau einer dieser sechs Klassen an.

Die Tableauregeln lassen sich jetzt einfach angeben, für jede der sechs Formelklassen gibt es eine Regel: Für jede α-Formel α und jedes Präfix σ:

$$\frac{\sigma\alpha}{\begin{array}{c}\sigma\alpha_1\\ \sigma\alpha_2\end{array}}$$

Für jede β-Formel β und jedes Präfix σ:

$$\frac{\sigma\beta}{\sigma\beta_1 \text{ oder } \sigma\beta_2}$$

Für jede γ-Formel γ und jedes Präfix σ:

$$\frac{\sigma\gamma}{\sigma\gamma_0(x)}$$

wobei x eine neue freie Variable ist.

Für jede δ-Formel δ und jedes Präfix σ:

$$\frac{\sigma\delta}{\sigma\delta_0(f(y_1,\ldots,y_n)(x))}$$

wobei f ein neues Skolemfunktionszeichen ist.

Für jede ν-Formel ν und jedes Präfix σ:

$$\frac{\sigma\nu}{\sigma'\nu_0}$$

wobei σ' von σ aus zugänglich sein muß und auf dem Pfad, der durch diese Regel verlängert wird, schon vorkommt.

Für jede π-Formel π und jedes Präfix σ:

$$\frac{\sigma\pi}{\sigma'\pi_0}$$

wobei σ' eine **einfache unbeschränkte Erweiterung** von σ, d.h. wenn σ' von σ aus erreichbar ist und σ' kein Anfangsstück eines anderen Präfix auf dem Pfad ist. Die angegebenen ν- und π-Regeln sind korrekt und vollständig für die modale Logik **K**. Für andere Logiken müssen neben einer möglichen Änderung der Zugänglichkeitsrelation für Präfixe auch diese beiden Regeln eventuell modifiziert werden. Ein Tableau, das mit Hilfe der obigen Regeln aufgebaut ist, nennen wir ein **K**-Tableau.

Ein **K**-Tableau T heißt geschlossen, wenn jeder Pfad von T ein Paar von Formeln der Form σFA und σTA enthält.

Beispiel 16 *(im S4-Kalkül)*

$$\begin{array}{rll}
1 & F\Box A \rightarrow \Box(\Box A \vee B) & (1) \\
1 & T\Box A & (2)\ \textit{aus}\ 1 \\
1 & F\Box(\Box A \vee B) & (3)\ \textit{aus}\ 1 \\
1,1 & F\Box A \vee B & (4)\ \textit{aus}\ 3 \\
1,1 & F\Box A & (5)\ \textit{aus}\ 4 \\
1,1 & FB & (6)\ \textit{aus}\ 4 \\
1,1,1 & FA & (7)\ \textit{aus}\ 5 \\
1,1,1 & TA & (8)\ \textit{aus}\ 2
\end{array}$$

Im **K***-Kalkül kann in Zeile 8 nur* $1,1\ TA$ *stehen, was nicht reicht, um das Tableau zu schließen.*

Beispiel 17 *(Im* **K***-Kalkül)*

$$\begin{array}{rll}
1 & F\Box A \rightarrow \Diamond A & (1) \\
1 & T\Box A & (2)\ \textit{aus}\ 1 \\
1 & F\Diamond A & (3)\ \textit{aus}\ 1
\end{array}$$

Die Formeln (2) und (3) sind beides ν*-Formeln. Es ist keine Regel mehr anwendbar. Würden wir allerdings die Einschränkung in der* ν*-Regel, daß nur ein zugängliches Präfix benutzt werden darf, das schon auf dem Pfad vorkommt, fallen lasssen, dann könnten wir das Tableau fortsetzen zu*

$$\begin{array}{rll}
1 & F\Box A \rightarrow \Diamond A & (1) \\
1 & T\Box A & (2)\ \textit{aus}\ 1 \\
1 & F\Diamond A & (3)\ \textit{aus}\ 1 \\
1,1 & TA & (4)\ \textit{aus}\ 2 \\
1,1 & FA & (5)\ \textit{aus}\ 3
\end{array}$$

und sogar schließen. Die Formel $\Box A \rightarrow \Diamond A$ *ist aber keine K-Tautologie. Fazit: Die Einschränkung an die* ν*-Regel ist notwendig für die Korrektheit des Kalküls.*

Auch an das Präfix, das bei der Anwendung der π-Regel benutzt wird, werden Einschränkungen geknüpft. Es muß neu sein, das ist klar, aber es darf auch nicht echtes Anfangsstück eines schon vorhandenen Präfix sein, dazu schaue man sich die Übungsaufgabe 3.7.1 an

Beispiel 18

1	$F(\Box A \wedge \Box B) \rightarrow \Box(A \wedge B)$	(1)
1	$T\Box A \wedge \Box B$	(2) *aus* 1
1	$F\Box(A \wedge B)$	(3) *aus* 1
1	$T\Box A$	(4) *aus* 2
1	$T\Box B$	(5) *aus* 2
1, 1	$FA \wedge B$	(6) *aus* 3
1, 1	TA	(7) *aus* 4
1, 1	TB	(8) *aus* 5

1, 1 FA (9) *aus* (6)	1, 1 FB (10) *aus* (6)
abgeschl. (9, 7)	*abgeschl.* (10, 8)

Beispiel 19

1	$F\Box(A \wedge B) \rightarrow (\Box A \wedge \Box B)$	(1)
1	$T\Box(A \wedge B)$	(2) *aus* 1
1	$F\Box A \wedge \Box B$	(3) *aus* 1

1	$F\Box A$	(4) *aus* 3	1	$F\Box B$	(5) *aus* 3
1, 1	FA	(6) *aus* 4	1, 1	FB	(10) *aus* 5
1, 1	$TA \wedge B$	(7) *aus* 2	1, 1	$TA \wedge B$	(11) *aus* 2
1, 1	TA	(8) *aus* 7	1, 1	TA	(12) *aus* 11
1, 1	TB	(9) *aus* 7	1, 1	TB	(13) *aus* 11
	abgeschl.	(8, 6)		*abgeschl.*	(13, 10)

Man beachte: Formel (2) wird zweimal benutzt.

Beispiel 20

1	$F\Box(A \vee B) \rightarrow (\Box A \vee \Box B)$	(1)
1	$T\Box(A \vee B)$	(2) *aus* 1
1	$F\Box A \vee \Box B$	(3) *aus* 1
1	$F\Box A$	(4) *aus* 3
1	$F\Box B$	(5) *aus* 3
1, 1	FA	(6) *aus* 4
1, 2	FB	(7) *aus* 5
1, 1	$TA \vee B$	(8) *aus* 2
1, 2	$TA \vee B$	(9) *aus* 2

$1,1\ TA \quad (10)[8]$	$1,1\ TB \quad (11)[8]$	
	$1,2\ TA \quad (12)[9]$	$1,2\ TB \quad (13)[9]$
abgeschl. $(10,6)$	$\vdots$	*abgeschl.* $(13,7)$
	offen	

Definition 75 *Ein Tableau T heißt* **K-erfüllbar**, *wenn es einen Pfad P in T, eine* **K**-*Kripkestruktur $(G, R, \models)$ und eine Abbildung I von der Menge aller Präfixe in P in die Menge G gibt, so daß*

$$I(\sigma)RI(\sigma') \quad \text{gilt, falls } \sigma' \text{ von } \sigma \text{ aus zugänglich ist,}$$

und

$$I(\sigma) \models A \quad \text{gilt für jede Formel } \sigma A \text{ auf dem Pfad } P.$$

Satz 67 (Korrektheitssatz)
Gibt es ein geschlossenes **K**-*Tableau mit Wurzelformel $1\ FB$, dann ist B in allen* **K**-*Kripkemodellen gültig.*

Beweis:
Mit T bezeichnen wir das gegebene geschlossene **K**-Tableau. Angenommen es gäbe eine **K**-Kripke-Struktur $(G, R, \models)$ und eine Welt $g \in G$ mit $g \models \neg B$. Der Wurzelknoten von T, $1FB$, ist dann für sich betrachtet ein **K**-erfüllbares Tableau, wenn wir die Abbildung I durch $I(1) = g$ definieren. Aus dem nachfolgenden Korrektheitslemma folgt, daß dann auch das ganze Tableau T **K**-erfüllbar sein müßte im Widerspruch zur Geschlossenheit von T.

■

Lemma 68 (Korrektheitslemma)
Ist T_0 ein **K**-*erfüllbares Tableau und T entstehe aus T_0 durch eine Regelanwendung, dann ist T ebenfalls* **K**-*erfüllbar.*

Beweis:
Wir betrachten nur die ν- und π-Regeln.

Kommt die Formel $\sigma\nu$ auf dem Pfad P vor und wird der Pfad P verlängert zu $P \cup \{\sigma'\nu_0\}$, wobei σ' schon als Vorzeichen auf P vorkommt und von σ aus erreichbar ist, so folgt nach Voraussetzung sowohl

$$I(\sigma)RI(\sigma')$$

als auch

$$I(\sigma) \models \nu.$$

Daraus folgt unmittelbar $I(\sigma') \models \nu_0$. (Man denke dabei zuerst an den typischen Fall $\nu = \Box A$.)

Kommt die Formel $\sigma\pi$ auf dem Pfad P vor und wird P verlängert zu $P \cup \{\sigma'\pi_0\}$, wobei σ' eine einfache, uneingeschränkte Erweiterung von σ ist.

Nach Voraussetzung gilt $I(\sigma) \models \pi$. Aus der Form von π-Formeln (man denke an die typische Form $\pi = T\Diamond\pi_0$) folgt die Existenz einer Welt $g \in G$, so daß

$$g \models \pi_0.$$

Wir setzen die Abbildung I fort durch

$$I(\sigma') = g.$$

Da σ' eine einfache Erweiterung von σ ist, gilt für jedes Präfix ρ auf dem Pfad:

$$\sigma' \text{ ist zugänglich von } \rho \qquad \text{gdw.} \qquad \rho = \sigma$$

Da σ' außerdem eine uneingeschränkte Erweiterung von σ ist, gilt für jedes Präfix ρ auf dem Pfad:

$$\rho \text{ ist nicht zugänglich von } \sigma' \text{ aus.}$$

Beide Eigenschaften zusammen sichern, daß für zwei beliebige Präfixe ρ_1, ρ_2 auf dem Pfad $P \cup \{\sigma'\pi_0\}$ gilt:

$$I(\rho_1)RI(\rho_2) \qquad \text{falls } \rho_2 \text{ ist zugänglich von } \rho_1 \text{ aus.}$$

Damit ist die Erfüllbarkeit von T' nachgewiesen.

∎

Satz 69 (K-Vollständigkeitssatz)
Ist A eine **K***-allgemeingültige Formel, dann gibt es ein geschlossenes* **K***-Tableau mit Wurzel* $1 \; FA$.

Beweis:
Der globale Aufbau des Beweises ist derselbe wie für die dreiwertige Logik, so daß wir schnell zum Kern der Konstruktion vordringen können. Sei P_0 die Menge der signierten Formeln mit Präfixen auf einem offenen erschöpften

Pfad. Insbesondere liegt für jede ν-Formel $\sigma\nu$ in P_0 auch $\sigma'\nu_0$ in P_0 für **jedes** Präfix σ', das in P_0 vorkommt und von σ aus erreichbar ist.

P entstehe aus P_0, indem die freien Variablen durch variablenfreie Terme ersetzt werden in der Weise, wie es im Beweis des Vollständigkeitssatzes für den dreiwertigen Tableaukalkül beschrieben ist. Insbesondere gilt für jede Formel $\sigma T \forall x A(x) \in P$ auch $\sigma T A(t/x) \in P$ für jeden variablenfreien Term t.

Gesucht ist ein **K**-Kripkemodell für P.

Die Menge G aller Welten sei die Menge aller Präfixe, die in P vorkommen; die Relation R sei die Zugänglichkeitsrelation auf der Menge aller Präfixe. Jedem Präfix (d. h. jeder Welt) wird dieselbe Prästruktur zugeordnet, nämlich die Herbrandstruktur in dem Vokabular einschließlich der bei Anwendungen der δ-Regel erzeugten Skolemfunktionen. Die Definition der **K**-Kripkestruktur wird vervollständigt durch die Festsetzung, daß eine atomare, variablenfreie Formel A in der Struktur, die zu einer Welt σ gehört, genau dann wahr sein soll, wenn $\sigma T A$ in P vorkommt.

Durch Induktion über den Aufbau signierter Formeln zeigt man, daß für jedes σC in P

$$\sigma \models C$$

in der Kripkestruktur gilt.

Wir beschränken uns auf zwei typische Beispiele.

Liegt $\sigma_0 T \Box B$ in P, dann liegt für jedes Präfix $\sigma \in G$, das von σ_0 aus erreichbar ist, auch $\sigma T B$ in P. Nach Induktionsvoraussetzung gilt somit für alle σ mit $\sigma_0 R \sigma : \ \sigma \models B$, woraus unmittelbar $\sigma_0 \models \Box B$ folgt.

Liegt $\sigma_0 F \Box B$ in P, so gibt es, weil wir von einem erschöpften Pfad ausgegangen sind, ein Präfix σ, das von σ_0 aus erreichbar ist, so daß $\sigma F B$ in P liegt. Nach Induktionsvoraussetzung gilt

$$\sigma \models F B$$

und damit auch

$$\sigma_0 \models F \Box B.$$

∎

Betrachten wir als Beispiel dafür, wie der **K**-Tableaukalkül variiert werden kann, den $S4$-Tableaukalkül. Ein Präfix σ_1 heißt $S4$-zugänglich von σ, falls σ

ein Anfangsstück von σ_1 ist. Der Fall $\sigma_1 = \sigma$ ist dabei zugelassen. Die Regeln bleiben unverändert.

Korrektheit und Vollständigkeit
Eine Formel A ist eine $S4$-Tautologie genau dann, wenn es ein geschlossenes $S4$-Tableau für $1\ FA$ gibt.

Bisher haben wir nur die Ableitbarkeit von Tautologien behandelt. Wesentlich interessanter ist jedoch die Frage, welche logischen Konsequenzen aus einer Menge Σ von Voraussetzungen folgen. Wie wir in der Einführung in die modale Logik schon gesehen haben gibt es zwei Varianten, die lokale und die globale logische Folgerungsbeziehung. Für beide wollen wir einen vollständigen und korrekten Tableaukalkül angeben, der glücklicherweise in beiden Fällen eine leichte Modifikationen des gerade vorgestellten Kalküls ist.

Definition 76 (L-Tableau)
Sei Σ eine Menge modallogischer Formeln.
Wir nennen ein K-Tableau T ein K-L-Tableau über Σ, falls beim Aufbau von T die folgende Regel benutzt werden konnte:

$$\frac{}{1TA} \qquad \textit{für jedes } A \in \Sigma$$

Theorem 70 (L-Vollständigkeitssatz)
Sei Σ eine Menge modallogischer Formeln, A eine modallogische Formel.
$\Sigma \vdash_L A$ gdw ein geschlossenes K-L-Tableau mit Wurzel $1FA$ existiert.

Definition 77 (G-Tableau)
Sei Σ eine Menge modallogischer Formeln.
Wir nennen ein K-Tableau T ein K-G-Tableau über Σ, falls beim Aufbau von T die folgende Regel benutzt werden konnte:

$$\frac{}{\sigma TA} \qquad \textit{für jedes } A \in \Sigma \textit{ und jedes Präfix } \sigma \textit{ auf dem Pfad}$$

Theorem 71 (G-Vollständigkeitssatz)
Sei Σ eine Menge modallogischer Formeln, A eine modallogische Formel.
$\Sigma \vdash_G A$ gdw ein geschlossenes K-G-Tableau mit Wurzel $1FA$ existiert.

Es bleibt noch ein Thema anzuspechen, das wieder einen Unterschied zwischen modaler Aussagenlogik und der klassischen Aussagenlogik aufzeigt, die Entscheidbarkeit.

Beispiel 21 *(im S4-Kalkül)*

1	$F\Box\Diamond p \rightarrow \Diamond\Box p$	(1)
1	$T\Box\Diamond p$	(2)[1]
1	$F\Diamond\Box p$	(3)[1]
1	$T\Diamond p$	(4)[2]
1	$F\Box p$	(5)[3]
1.1	Tp	(6)[4]
1.2	Fp	(7)[5]
1.1	$T\Diamond p$	(8)[2]
1.1	$F\Box p$	(9)[3]
1.2	$T\Diamond p$	(10)[2]
1.2	$F\Box p$	(11)[3]
1.1.1	Tp	(12)[8]
1.1.2	Fp	(13)[9]
1.2.1	Tp	(14)[10]
1.2.2	Fp	(15)[11]

Offensichtlich terminiert die Tableaukonstruktion in diesem Beispiel nicht. Ein geschlossenen Tableau kann es auch nicht geben, denn $\Box\Diamond p \rightarrow \Diamond\Box p$ ist keine S4-Tautologie. Aber im Unterschied zu klassischen aussagenlogischen Tableaus muß in der modalen Aussagenlogik der Beweisversuch einer Nichttautologie nicht terminieren. Das dieses Phänomen in der Logik **K** nicht auftreten kann, wird in Aufgabe 3.7.3 thematisiert. Es wurden jedoch Modifikationen des Kalküls vorgeschlagen, die dieses Defizit beheben und damit zeigen, daß die normalen modalen Aussagenlogiken entscheidbar sind, siehe auch Übugnsaufgabe 3.7.4.

3.7.2 Tableaukalküle mit Präfixvariablen

Ein Nachteil der in Abschnitt 3.7 vorgestellen Tableaukalküle für modale Logiken liegt in der speziellen Form der ν-Regeln: in jeder Regelanwendung muß ein festes Präfix gewählt werden, obwohl nicht immer klar ist, welche Wahl letztendlich zur Erzeugung eines komplementären Paares beiträgt. Die Situation ist vergleichbar mit der in [Smullyan, 1968] benutzten γ-Regel, die eine feste variablenfreie Instanziierung für universell quantifizierte Variablen fordert. Wir stellen in diesem Abschnitt einen Kalkül mit einer ν-Regel vor, die freie Variablen im Präfix benutzt. Wir führen zunächst den einfachen Fall eines Tableaukalküls für die minimale normale Logik **K** vor. Die Anpassung an andere modale Logiken ist nicht immer trivial.

Definition 78 (Präfix) *Ein Präfix t ist eine endliche Folge bestehend aus natürlichen Zahlen und Präfixvariablen, die wir mit Großbuchstaben aus dem Ende des Alphabets bezeichnen, U, V, W. Formal wird die Menge, $\mathcal{S}$, aller Präfixe induktiv definiert:*

1. *die leere Folge [] ist in $\mathcal{S}$,*
2. *für $\sigma \in \mathcal{S}$ und $n \in I\!N$ gilt auch $\sigma n \in \mathcal{S}$,*
3. *ist $\sigma \in \mathcal{S}$ und U eine Präfixvariable, dann gilt auch $\sigma U \in \mathcal{S}$.*

Ein Präfix ohne Präfixvariable heißt ein **variabelenfreies Präfix**. *Wir bezeichnen mit $SVar$ die Menge aller Präfixvariablen.*
Eine Präfixformel ist von der Form $\sigma\ A$ mit $A \in Fml_{ALmod}$ und $\sigma \in \mathcal{S}$.

Eine Funktion $\Phi : SVar \rightarrow I\!N^*$ XS heißt eine Präfixsubstitution. Wenn keine Verwechslung zu befürchten ist sprechen wir einfach von einer Substitution ohne weiteren Zusatz. Die Eigenschaften der Subsitution Φ hängen von der betrachteten Logik ab. Für die Logik **K** muß z.B. für jede Variable $X \in SVar$ die Folge $\Phi(X)$ einelementig sein. Das Ergebnis der Anwendung einer Präfixsubstitution Φ auf ein Präfix σ bezeichnen wir wie üblich mit $\Phi(\sigma)$. Eine Substitution Φ heißt ein Unifikator der beiden Präfixe σ und τ falls $\Phi(\sigma) = \Phi(\tau)$ gilt. In diesem Fall heißen σ und τ unifizierbar.

Die Tableaus, die wir im folgenden betrachten werden, fassen wir wie üblich als Menge von Pfaden auf, wobei ein Pfad seinerseits eine Menge von

Präfixformeln ist. Für einen Pfad B bezeichnen wir mit $pre(B)$ die Menge der in B vorkommenden Präfixe, $gp(B)$ steht für die Menge der variablenfreien Präfixe, die in B vorkommen. Es wird sich außerdem als hilfreich erweisen die Hilfsfunktion $gp(\sigma, B)$ zur Verfügung zu haben, die definiert ist durch:

$$gp(\sigma, B) = \{\Phi(\sigma) \mid \text{ mit } \Phi(\sigma) \in gp(B)\}$$

Der Kalkül für die minimale normale Logik **K** enthält die üblichen Tableauregeln für die aussagenlogischen Junktoren und Quantoren und die folgenden

Box Regel	**Diamant-Regel**
$\dfrac{\sigma\ \Box A}{\sigma U\ A}$	$\dfrac{\sigma\ \Diamond A}{\sigma k\ A}$

Wird die Box-Regel auf einem Pfad B benutzt, dann muß die Präfixvariable U neu sein, d.h. sie darf nicht schon in $pre(B)$ vorkommen. Wird die Diamant-Regel auf dem Pfad B angewandt, dann muß $k \in I\!N$ so gewählt werden, daß $\sigma' k \notin gp(B)$ gilt für alle $\sigma' \in gp(\sigma, B)$.
Eine Substitution μ und eine Präfixsubstitution Φ schließen einen Pfad B ab, wenn es Präfixformeln $\sigma\ A$ und $\tau\ \neg C$ in B gibt mit $\Phi(\sigma) = \Phi(\tau)$ und $\Phi(\sigma) \in gp(B)$ und $\mu(A) = \mu(C)$. Die Forderung, daß τ und σ unifizierbar sind allein würde zu einem inkorrekten Kalkül führen, wie das Beispiel 22 zeigt. Ein Tableau heißt geschlossen, wenn es Substitutionen Φ und μ gibt, die alle Pfade schließen.

Beispiel 22
Wir betrachten die erfüllbare Formelmenge $M = \{\Box\Diamond p, \Box\Diamond\neg p\}$. Durch Erweiterung des initialen Tableau für M erreicht man einen Pfad B, der die Präfixformeln $1.X.1\ p$ und $1.Y.1\ \neg p$ enthält. Die Folgen $1.X.1$ and $1.Y.1$ sind unifizierbar, aber für kein Φ gilt $\Phi(1.X.1) \in gp(B)$.

Jede Formel, mit Ausnahme der ν-Formeln, braucht auf jedem Pfad nur einmal benutzt zu werden.

Definition 79
Ein Tableau T heißt erfüllbar, wenn es eine Kripke Struktur $\mathcal{K} = (G, R, M)$, gibt, so daß für jede Belegung β der freien Variablen in T ein Pfad B existiert und eine Abbildung $\mu : gp(B) \to G$ mit

1. *für jede Präfixformel $\sigma\ A$ in B und alle $\sigma' \in gp(\sigma, B)$ gilt $(\mathcal{K}, \mu(\sigma')) \models A[\beta]$,*

2. *für zwei Folgen $\sigma \in gp(B)$ und $\sigma k \in gp(B)$ gilt $\mu(\sigma) R \mu(\sigma k)$.*

Für $M = \{A_1, \ldots, A_j\} \subseteq Fml_{ALmod}$ besteht das **initiale Tableau** für M aus einem Pfad mit den Knoten $n_1, \ldots, n_j$ wobei n_i für alle $1 \leq i \leq j$ die Markierung $1\ A_i$ trägt.

Lemma 72 *Falls das initiale Tableau für M erfüllbar ist, dann ist auch M erfüllbar.*

Proof: Offensichtlich.

Lemma 73 *Sei T ein erfüllbares Tableau und T_1 entstehe aus T durch Anwendung einer Tableauregel. Dann ist T_1 auch erfüllbar.*

Beweis: Der Beweis benutzt Fallunterscheidung nach dem Typ der Tableauregel, die benutzt wurde um T_1 aus T zu erreichen. Wir führen hier nur die kritischen Fälle der Box- und Diamant-Regel vor.
Sei B ein Pfad in T, $\sigma\ \Box A$ eine Formel auf B und T_1 entstehe aus T durch die Erweiterung des Pfades B um die Vorzeichenformel $\sigma U\ A$.
Da T als erfüllbar vorausgesetzt war gibt es eine Kripke Struktur $\mathcal{K} = (G, R, v)$, einen Pfad B_1 von T und ein Abbildung $\mu : gp(B_1) \to G$, so daß die Bedingungen aus Definition 79 erfüllt sind. Offensichtlich müssen wir nur den Fall $B = B_1$ betrachten, da sonst T_1 trivialerweise erfüllbar bleibt. Um die Erfüllbarkeit von T_1 zu zeigen, benutzen wir dieselbe Kripke Struktur $\mathcal{K}$ und den Pfad $B \cup \{\sigma U\ A\}$. Da $gp(B \cup \{\sigma U\ A\}) = gp(B)$ gilt können wir auch dieselbe Abbildung μ benutzen. Nach Voraussetzung gilt für alle $\sigma' \in gp(\sigma, B)$ $(\mathcal{K}, \mu(\sigma')) \models \Box A$. Offensichtlich ist jeder Präfix $\tau \in gp(\sigma U, B)$ von der Form $\sigma' k$ für ein $\sigma' \in gp(\sigma, B)$ und ein $k \in I\!N$. Nach Voraussetzung über μ gilt $\mu(\sigma') R \mu(\sigma' k)$ woraus unmittelbar $(\mathcal{K}, \mu(\sigma' k)) \models A$ folgt.
Betrachten wir jetzt den Fall, daß T_1 durch Anwendung der Diamantregel auf die Formel $\sigma\ \Diamond A$ in B entsteht. In diesem Fall wird B zu $B \cup \{\sigma k\ A\}$ erweitert. Wir benutzen dieselbe Notation wie im vorangegangenen Fall. Um die Erfüllbarkeit von T_1 zu beweisen benutzen wir wieder dieselbe Kripke Struktur $\mathcal{K}$ und den Pfad $B \cup \{\sigma k\ A\}$. Wegen $gp(B \cup \{\sigma k A\}) = gp(B) \cup \{\sigma' k \mid \sigma' \in gp(\sigma, B)\}$ müssen wir die Funktion μ an den Argumentstellen $\sigma' k$ definieren.

Nach Vorasusetzung gilt für jedes $\sigma' \in gp(\sigma, B)$ jedenfalls $(\mathcal{K}, \mu(\sigma')) \models \diamond A$. Es gibt also ein $g \in G$ mit $(\mathcal{K}, g) \models A$. Wir wählen ein solches g und definieren $\mu(\sigma' k) = g$. Woraus unmittelbar die Erfüllbarkeit von T_1 folgt. Es ist wichtig an dieser Stelle zu bemerken, daß die durch die π-Regel verlangte Einschränkung an k sicherstellt, daß $\sigma' k \notin gp(B)$ für alle $\sigma' \in gp(\sigma, B)$ gilt.

▪

Theorem 74
Sei $M \subseteq Fml_{ALmod}$. Falls das initiale Tableau T_0 für M sich zu einem geschlossenen Tableau T erweitern läßt, ist M nicht erfüllbar.

Proof: Wir wollen uns zunächst davon überzeugen, daß das Tableau T nicht erfüllbar ist. Anderenfalls gäbe es eine Kripke Struktur $\mathcal{K} = (G, R, v)$, einen Pfad B von T und eine Abbildung $\mu : gp(B) \to G$, so daß die Bedingungen aus Definition 79 erfüllt sind. Sei Φ die Präfixsubstitution, die zum Abschluß von T benutzt wurde. Da Φ insbesondere den Pfad B schließen muß gibt es Formeln $\tau\ A$ and $\sigma\ \neg A$ auf B mit $\Phi(\sigma) = \Phi(\tau) \in gp(B)$. Das führt aber zu dem Widerspruch $(\mathcal{K}, g) \models A \wedge \neg A$ for $\mu(\Phi(\sigma)) = \mu(\Phi(\tau)) = g$.
Mit Lemma 72 folgt aus der Unerfüllbarkeit von T die Unerfüllbarkeit von T_0 und daraus die Unerfüllbarkeit von M.

▪

Theorem 75 (Vollständigkeit)
Wenn $M \subseteq Fml_{ALmod}$ nicht erfüllbar ist, dann kann das initiale Tableau T_0 für M zu einem geschlossenen Tableau erweitert werden.

Beweis: Da der quantorenlogische Teil nicht in kritischer Weise mit dem modallogischen Teil interagiert, beschränken wir uns in diesem Beweis auf die Aussagenlogik und überlassen es dem Leser, sich die Erweiterung auf die Prädikatenlogik zu überlegen.
Wir nehmen an, daß es keine geschlossene Erweiterung von T_0 gibt. Daraus schließen wir auf die Existenz eines offenen Pfades P, mit den folgenden Eigenschaften:

1. für jede Formel $A \in M$ gilt $1\ T A \in P$,
2. für jedes α-Formel $\sigma\ \alpha \in P$ gilt $\sigma\ \alpha_1 \in P$ und $\sigma\ \alpha_2 \in P$,

3. für jedes β-Formel $\sigma\ \beta \in P$ gilt $\sigma\ \beta_1 \in P$ oder $\sigma\ \beta_2 \in P$,

4. für jede ν-Formel $\sigma\ \nu \in P$ gilt für jede Präfixvariable $U \in PVar$ auch $\sigma U\ \nu_0 \in P$,

5. für jede π-Formel $\sigma\ \pi \in P$ gibt es ein $k \in N$ mit $\sigma k\ \pi_0 \in P$.

Sei Φ eine Präfixsubstitution, so daß für alle $\sigma\ A \in P$ gilt $\Phi(\sigma) \in gp(P)$. Wir setzen $P^* = \Phi(P)$. Außerdem sei Φ so gewählt, daß für alle $\sigma U\ A \in P$ und alle $\sigma' \in gp(\sigma U, P)$ auch $\sigma'\ A \in P^*$ gilt. Das ist möglich wegen Bedingung 4 an P. Wir konstruieren aus P^* eine Kripke-Struktur

$$\begin{array}{lcl}
\mathcal{K} & = & (G, R, M) \\
G & = & gp(B) \\
sRt & \Leftrightarrow & length(t) = length(s) + 1 \\
v(\sigma, p) = 1 & \Leftrightarrow & \sigma\ p \in P^*
\end{array}$$

Jetzt kann man leicht beweisen, durch Induktion über die Komplexität von A, daß für jede Präfixformel $(\sigma\ A) \in P^*$ gilt $(\mathcal{K}, \sigma) \models A$.

$\blacksquare$

3.7.3 Übungsaufgaben

Übungsaufgabe 3.7.1 *Zeigen Sie an einem Gegenbeispiel, daß die Einschränkung bzgl. des neu zu wählenden Präfix in der π-Regel notwendig ist für die Korrektheit des Kalküls.*

Übungsaufgabe 3.7.2 *Wo im Beweis von Satz 67 wird die Einschränkung an die Anwendung der π-Regel benutzt*

Übungsaufgabe 3.7.3 *In dieser Aufgabe betrachten wir nur aussagenlogische Formeln und Kalküle. Die Länge einer aussagenlogischen Formel A, bezeichnet mit* laenge(A) *ist die Anzahl der Symbole in A, wobei Klammern nicht mitgezählt werden.*

Definition 80
Zu jedem Pfad P eines Tableau und jedem Präfix σ definieren wir die folgende Menge von Vorzeichenformeln

$$F_\sigma(P) = \{A \mid \sigma A \in P\}$$

Ist P eine Formelmenge $P \subseteq Fml_{ALmod}$ dann bezeichnen wir mit $subfml(P)$ die Menge aller Teilformel von Formeln aus P. Ist P eine Menge signierter Formeln, so soll auch $subfml(P)$ die Menge aller signierten Teilformeln von Formeln aus P bezeichnen. Begründen Sie die Richtigkeit der folgenden Aussagen:

1. *Ist P ein Ast in einem Tableau mit der Wurzelmarkierung $1FA$, dann gilt für alle Präfixe σ*

$$\mathrm{subfml}(F_\sigma(P)) \subseteq \mathrm{subfml}(FA)$$

2. *Ist P ein Pfad eines Tableaux im **K**-Kalkül und ist das Präfix σ_2 von σ_1 erreichbar, dann gilt*

$$\max(\mathrm{laenge}(A) \mid A \in F_{\sigma_2}(P)) < \max(\mathrm{laenge}(A) \mid A \in F_{\sigma_1}(P))$$

Übungsaufgabe 3.7.4

*Diese Aufgabe soll eine Modifikation des **S4**-Tableaukalküls vorstellen, die für alle aussagenlogische Formeln terminiert.*

*Als erstes erweist es sich als nützlich und für den angestrebten Vollständigkeitsbeweis sogar als notwendig, die ν-Regel zur **S4**-ν-Regel abzuändern.*

Allgemeine Form	*Instanz*
$\dfrac{\sigma\ \nu}{\begin{array}{c}\sigma'\ \nu_0\\ \sigma'\ \nu\end{array}}$	$\dfrac{\sigma\ \Box\phi}{\begin{array}{c}\sigma'\ \phi\\ \sigma'\ \Box\phi\end{array}}$

Der Unterschied besteht darin, daß die ν-Formel in die neue Welt *mit hinübergenommen wird. Die Einschränkungen an σ' sind dieselben wie vorher.*

1. *Die erste Teilaufgabe besteht in dem Nachweis, daß die **S4**-ν-Regel korrekt ist.*

Zur Formulierung der Aufgabe brauchen wir noch etwas neue Terminologie

Definition 81
Sei P ein Pfad in einem Tableau für eine modale Aussagenlogik. Der Präfixbaum *von P ist die Menge aller Präfixe, die auf P vorkommen, mit der Zugänglichkeitsrelation als Ordnungsrelation.*

Wir vereinbaren die folgende Notation für Anfangsstücke von Präfixen. Ist $\sigma = n_1 \ldots n_k$ ein Präfix der Länge k, dann steht $\sigma^j = n_1 \ldots n_j$ mit $1 \leq j \leq k$ für das Anfangstück der Länge j von σ.

Wir wiederholen noch einmal die Definition der Zugänglichkeit in diesem Fall. Päfix σ_2 ist von σ_1 aus zugänglich, in Zeichen $\sigma_1 < \sigma_2$, wenn σ_1 ein echtes Anfangsstück von σ_2 ist. Die Änderung besteht in einer Abbruchbedingung, die wir gleich formulieren werden. Der Abbruch darf jedoch nicht zu früh erfolgen. Was das genau bedeutet wollen wir zunächst präzisieren.

Definition 82
Sei P der Ast eines Tableaux.

1. *Ein Präfix σ heißt ein* Wiederholungpräfix *auf P, wenn für jede ν-Formel $T\Box\nu_0$ oder $F\Diamond\nu_0$ in $F_\sigma(P)$ auch $T\nu_0$ bzw $F\nu_0$ in $F_\sigma(P)$ liegt.*

2. *Zwei Präfixe σ_1 und σ_2 wollen wir als äquivalent ansehen, in Zeichen $\sigma_1 \sim \sigma_2$, wenn*
   - *$F_{\sigma_1}(P) = F_{\sigma_2}(P)$ und*
   - *es gibt Anfangsstücke σ'_1 von σ_1 und σ'_2 von σ_2, nicht notwendigerweise echte, die beide Wiederholungpräfixe sind.*

3. *Für zwei Präfixe σ_1 und σ_2 trifft die Relation $R_\sim$ zu, wenn es eine Folge von Präfixen ρ_1, ρ'_1, $\ldots \rho_k$, ρ'_k gibt, so daß gilt*
   - *$\rho_1 = \sigma_1$ und $\rho'_k = \sigma_2$*
   - *für alle i gilt $\rho_i \sim \rho'_i$*
   - *für alle i ist ρ'_i ein Anfangsstück von ρ_{i+1}*

Die Anwendung der Tableauregeln erfolgt wie bisher. Das Verfahren terminiert, wenn ein geschlossenes Tableau erreicht wird oder, und das ist das Neue, wenn es einen offenen Pfad P gibt, der die folgende Abbruchbedingung erfüllt:

1. *außer π-Regeln ist keine Regel auf P anwendbar*

2. *ist σ kein Blatt im Präfixbaum, dann wurde für* **jede** *π-Formel in $F_\sigma(P)$ die entsprechende π-Regel angewendet.*

3. *für jedes Blatt σ des Präfixbaums von P, so daß $F_\sigma(P)$ eine π-Formel enthält, gibt es ein Präfix σ' mit $R_\sim(\sigma, \sigma')$ und $\sigma \not\sim \sigma'$*

Zeigen Sie

2. *Dieses Verfahren terminiert immer*

3. *Trifft die Abbruchbedingung auf den Pfad P zu, so lässt sich aus P ein Modell für FA konstruieren.*
   Hinweis: Folgen Sie der Konstruktion im Beweis von Satz 71 wobei die Päfixe σ_1,σ_2 des Präfixbaums mit $\sigma_1 \sim \sigma_2$ identifiziert werden, d.h. nur **eine** *Kripkewelt liefern.*

Übungsaufgabe 3.7.5
Wenden Sie das in Aufgabe 3.7.4 beschriebene Verfahren mit der Wurzelmarkierung

$$\Box\Diamond p \land \Diamond p \to \Diamond\Box p$$

an.

3.8 Dynamische Logik

3.8.1 Einleitung

Die dynamische Logik liefert eine Sprache zur Beschreibung der Semantik von Programmiersprachen und einen Kalkül zur Verifikation von Programmen in diesen Programmiersprachen. Als Vorläufer dieses Ansatzes einer Logik für Programme wird üblicherweise die Arbeit von V. R. Pratt zitiert [V.R.Pratt, 1976]. Der Begriff *dynamische Logik* und die spezielle Form, die wir in diesem Kapitel präsentieren, stammt aus der Arbeit [Harel, 1979], die eine erweiterte Form der Dissertation von David Harel aus dem Jahr 1978 ist. Lesenswerte Referenzen sind auch heute noch [Harel, 1984], [Goldblatt, 1987] Section 10 und Part Three, [Goldblatt, 1982]. Die aktuellste Referenz ist das Buch [Harel *et al.*, 2000].

Die Syntax der dynamischen Logik ist eine Erweiterung, allerdings eine sehr weitreichende Erweiterung, der aus dem Hoare Kalkül bekannten Aussagen. Wir betrachten zum Einstieg noch einmal die wichtigsten Elemente des Hoareschen Ansatzes.

Im Hoaresche Ansatz zur Programmverifikation versucht man das Programm mit Vor- und Nachbedingungen zu annotieren, siehe Abbildung 3.8, die lokal bewiesen werden können und zusammengenommen die gewünschte Korrektheitsaussage ergeben.

Die Zuweisungsgruppe `x:=a;y:=b` hat als Vorbedingung $C0 \;\; a \geq 0 \wedge b \geq 0$. Die Nachbedingungen sind $C1 \;\; x = a \wedge y = b$ und

$$C2 \;\; \forall u((u \mid x \wedge u \mid y) \leftrightarrow (u \mid a \wedge u \mid b)).$$

Hierbei steht $\mid$ für die ist-Teiler Relation. Die beiden Nachbedingungen gelten offensichtlich ohne von der Vorbedingung Gebrauch machen zu müssen. Die Vorbedinung ist wichtig, um die Terminierung des Programms sicherzustellen. Aus Abbildung 3.8 liest man ferner ab, daß die Bedingung $C2$ auch nach jedem Schleifendurchlauf gelten soll. Das ist leicht einzusehen. Die Bedingung die nach Verlassen der Schleife gelten soll ist

$$\begin{array}{c} C3 \;\; (x = 0 \wedge \forall u(u \mid y \leftrightarrow (u \mid a \wedge u \mid b))) \\ \vee \\ (y = 0 \wedge \forall u(u \mid x \leftrightarrow (u \mid a \wedge u \mid b))). \end{array}$$

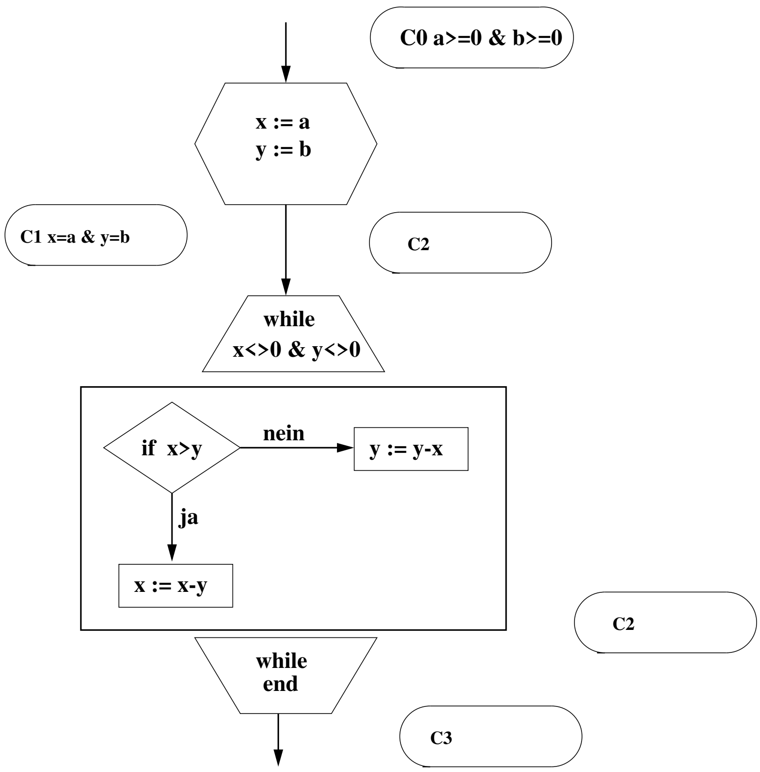


Abbildung 3.8: Flußdiagramm mit Annotationen

Wenn die Schleife verlassen wird muß die Schleifenbedingung verletzt sein, d.h. es gilt $x = 0 \lor y = 0$. Die Konjunktion von $C2$ mit dieser Formel liefert zunächst

$$\begin{array}{c} (x = 0 \land \forall u((u \mid x \land u \mid y) \leftrightarrow (u \mid a \land u \mid b))) \ , \\ \lor \\ (y = 0 \land \forall u((u \mid x \land u \mid y) \leftrightarrow (u \mid a \land u \mid b))). \end{array}$$

woraus wegen der Allgemeingültigkeit von $x \mid 0$ schließlich $C3$ folgt. Der Hoare-Kalkül stellt ein formales Regelwerk zur Verfügung, mit dem man die Schlüsse, die wir eben informal gezogen haben, streng formal und damit automatisierbar nachvollziehen kann.

3.8.2 Eine Dynamische Aussagenlogik für while-Programme

Wir betrachten zunächst die dynamische Aussagenlogik PDL_{while} und im nächsten Abschnitt PDL_{reg} und kommen erst später in Unterabschnitt 3.8.4 auf dynamische Quantorenlogiken zu sprechen.

Auf der aussagenlogischen Ebene treten keine Individuenvariablen und auch keine Programmvariablen auf. Es gibt daher auf dieser Ebene keine Möglichkeit die einfachsten Programmkonstrukte, nämlich die Zuweisungen, zu beschreiben. Statt dessen treten, ähnlich den aussagenlogischen Variablen in der normalen Aussagenlogik, atomare Programme, Π_0, auf. Wie in der Prädikatenlogik aussagenlogische Variablen durch Formeln ersetzt werden, werden in der dynamischen Quantorenlogik, siehe Abschnitt 3.8.4, atomare Programme durch konkrete Programmbefehle ersetzt.

Definition 83 (PDL_{while}-Formeln und -Programme) *Sei P eine Menge aussagenlogischer Variablen und Π_0 eine Menge atomarer Programme.*

1. *jede Variable $p \in P$ ist eine PDL_{while}-Formel,*
2. *sind A, B PDL_{while}-Formeln, dann auch die aussagenlogischen Kombinationen $A \land B$, $A \lor B$, $A \rightarrow B$ und $\neg A$,*
3. *jedes atomare Programm $\pi \in \Pi_0$ ist ein PDL_{while}-Programm,*
4. *$\underline{\text{skip}}$, $\underline{\text{abort}}$ sind PDL_{while}-Programme,*

5. *sind* π_1,π_2 PDL_{while}*-Programme, dann auch* $(\pi_1; \pi_2)$,

6. *sind* π, π_1 PDL_{while}*-Programme und* A *eine* PDL_{while}*-Formel, dann sind* $\underline{while}\ A\ \underline{do}\ \pi\underline{end}$ *und* $\underline{if}\ A\ \underline{then}\ \pi\ \underline{else}\ \pi_1\underline{end}$ PDL_{while}*-Programme,*

7. *ist* A *eine* PDL_{while}*-Formel und* π *ein* PDL_{while}*-Programm, dann sind* $[\pi]A$, $\langle\pi\rangle A$ PDL_{while}*-Formeln.*

Ein Programm bewirkt einen Übergang von einem Zustand w_1 zu einem Zustand w_2. Auf der aussagenlogischen Ebene ist die Menge der Zustände eine nicht weiter präzisierte Menge W. Die Interpretation $\rho(\pi)$ eines Programmes besteht aus der Angabe einer Menge von Paaren (s, t) von Zuständen, so daß π einen Zustandsübergang von s nach t bewirkt. Da auch indeterministische Programme in die Betrachtungen mit eingeschlossen werden sollen, muß $\rho(\pi)$ keine Funktion sein, sondern kann eine beliebige Relation zwischen Zuständen sein.

Definition 84 (DAL-Struktur)
Eine Struktur $\mathcal{A} = (W, \tau, \rho)$ *für die dynamische Aussagenlogik besteht aus*

- *einer nichtleeren Menge* W *von Zuständen,*
- *einer Belegung* τ *der aussagenlogischen Variablen mit Wahrheitswerten. Genauer ordnet* τ *jeder aussagenlogischen Variablen,* p*, die Menge von Zuständen zu, in denen* p *wahr ist, d.h. für* $p \in F_0$ *ist* $\tau(p) \subseteq W$*,*
- *einer Interpretation* ρ *der atomaren Programme, d.h. einer Funktion* $\rho : \Pi_0 \to W \times W$

Man beachte, daß eine DAL-Struktur nicht anderes ist, als eine Kripke Struktur für eine multi-modale Logik, die für jedes $p \in \Pi$ eine *Zugänglichkeitsrelation* $\rho(\pi)$ bereitstellt.

Definition 85 (Semantik für PDL_{while}**)**
Sei $\mathcal{A} = (W, \tau, \rho)$ *eine Struktur für die dynamische Aussagenlogik, dann wird die Funktion* τ *fortgesetzt auf alle* PDL_{while}*-Formeln und* ρ *fortgesetzt auf alle Programme aus* Π *durch die folgende simultane induktive Definition*

1. $\tau(\neg F) = W \setminus \tau(F)$

2. $\tau(F_1 \vee F_2) = \tau(F_1) \cup \tau(F_2)$

3. $\tau(\langle\alpha\rangle F) = \{s \in W \mid$ *es gibt ein* t *mit* $(s,t) \in \rho(\alpha)$ *und* $t \in \tau(F)\}$

4. $\rho(\alpha;\beta) = \{(s,t) \mid$ *es gibt ein* u *mit* $(s,u) \in \rho(\alpha)$ *und* $(u,t) \in \rho(\beta)\}$

5. $(r,s) \in \rho(\underline{while}\ A\ \underline{do}\ \ \pi\underline{end}\ )$ *gdw es gibt* $r_0, \ldots, r_n \in W$, *sodaß* $r_0 = r$, $r_n = s$ *und für alle* i *mit* $0 \leq i < n$ *gilt* $(r_i, r_{i+1}) \in \rho(\pi)$ *und* $r_i \in \tau(A)$ *und* $r_n \notin \tau(A)$

6. $(r,s) \in \rho(\underline{if}\ A\ \underline{then}\ \pi\ \underline{else}\ \pi_1\underline{end}\ )$ *gdw* $r \in \tau(A)$ *und* $(r,s) \in \rho(\pi)$ *oder* $r \notin \tau(A)$ *und* $(r,s) \in \rho(\pi_1)$

Jetzt sollte sich eine Untersuchung der Eigenschaften von PDL_{while} anschließen. So möchte man z.B. wissen, ob PDL_{while} entscheidbar ist. Wir verschieben diese Fragen auf den nächsten Abschnitt. Dort wird eine dynamische Aussagenlogik PDL_{reg}eingeführt, die mit möglichst primitiven Programmkonstrukten arbeitet. Es stellt sich heraus, daß PDL_{while} in PDL_{reg}eingebettet werden kann. Aus diese Weise würde aus der Entscheidbarkeit von PDL_{reg}auch die Entscheidbarkeit von PDL_{while}folgen.

3.8.3 Eine minimale Dynamische Aussagenlogik

Ziel der dynamischen Logik war urspünglich nicht die direkte Anwendbarkeit in der Programmverifikation sondern am besten beschrieben durch das folgende Zitat aus [Harel, 1984]

> ... the idea is to study the mathematical properties of programs and their behaviour and the tools for reasoning about them, with the objective of singling out and gaining a better understanding of those concepts and methods which are fundamental to the reasoning process.

Verschiedene dynamische Aussagenlogiken unterscheiden sich in erster Linie durch die erlaubten Programme. Die in diesem Unterkapitel betrachtete dynamischen Logik erlaubt die Progamme Π_{min}, die aus Programmkonstrukte sehr abstrakter und allgemeiner Art aufgebaut sind, ganz im Einklang mit dem oben zitierten Ziel des ganzen Unterfangens.

Definition 86 (PDL_{reg}-Formeln und -Programme)
Die Menge der Formeln der dynamischen Aussagenlogik $Fml_{PDL_{reg}}$ und der erlaubten Programme Π wird simultan induktiv definiert durch

1. *die Konstanten* wahr *und* falsch *sind in $Fml_{PDL_{reg}}$,*
2. *jede aussagenlogische Variable ist in $Fml_{PDL_{reg}}$,*
   i.e. $F_0 \subset Fml_{PDL_{reg}}$,
3. *sind $F_1, F_2 \in Fml_{PDL_{reg}}$, dann auch $\neg F_1$, $(F_1 \vee F_2)$,*
4. *für $F \in Fml_{PDL_{reg}}$ und $\alpha \in \Pi$ gilt $<\alpha> F \in Fml_{PDL_{reg}}$,*
5. *jedes atomare Programm ist ein Programm, i.e. $\Pi_0 \subset \Pi$,*
6. *sind α, β in Π, dann auch $(\alpha;\beta)$, $(\alpha \cup \beta)$ und (α^*),*
7. *für $F \in Fml_{PDL_{reg}}$ gilt $(F?) \in \Pi$.*

Π_0	$=$	*Menge der atomaren Programme*
F_0	$=$	*Menge der aussagenlogischen Variablen*

Man beachte, daß weder $Fml_{PDL_{reg}}$ noch Π_{min} unabhängig voneinander definiert werden können.
Bevor wir gleich zu einer präzisen Definition der Semantik der eben definierten Formeln und Programme kommen, geben wir eine informelle Beschreibung der intendierten Bedeutung:

$<\alpha> F \in Fml_{PDL_{reg}}$	es gibt eine Ausführung von α nach der F gilt
$(\alpha;\beta)$	Hintereinanderausführung von α und β
$(\alpha \cup \beta)$	nichtdeterministische Auswahl zwischen α und β
(α^*)	indeterministisch endliche Wiederholung von α
$(F?) \in \Pi_{min}$	Fortsetzung nur wenn F wahr ist

Definition 87 (Semantik für PDL_{reg})
Sei $\mathcal{A} = (W, \tau, \rho)$ eine Struktur für die dynamische Aussagenlogik, dann wird die Funktion τ fortgesetzt auf alle Formeln aus $Fml_{PDL_{reg}}$ und ρ fortgesetzt auf alle Programme aus Π_{min} durch die folgende simultane induktive Definition

$[\alpha]F$	für	$\neg <\alpha> \neg F$
if A **then** α **else** β	für	$(A?;\alpha) \cup (\neg A?;\beta)$
while A **do** α	für	$(A?;\alpha)^*;\neg A?$
repeat α **until** A	für	$\alpha;(\neg A?;\alpha)^*$

Abbildung 3.9: Einige definierte Programmkonstrukte

1. $\tau(wahr) = W$ *und* $\tau(falsch) = \emptyset$
2. $\tau(\neg F) = W \setminus \tau(F)$
3. $\tau(F_1 \vee F_2) = \tau(F_1) \cup \tau(F_2)$
4. $\tau(<\alpha> F) = \{s \in W \mid$ *es gibt ein* t *mit* $(s,t) \in \rho(\alpha)$ *und* $t \in \tau(F)\}$
5. $\rho(\alpha;\beta) = \{(s,t) \mid$ *es gibt ein* u *mit* $(s,u) \in \rho(\alpha)$ *und* $(u,t) \in \rho(\beta)\}$
6. $\rho(\alpha \cup \beta) = \rho(\alpha) \cup \rho(\beta)$
7. $\rho(\alpha^*) = \{(s,t) \mid$ *es gibt ein* k *und es gibt* $s_0, \ldots, s_k$ *mit* $s_0 = s, s_k = t$ *und* $(s_{i-1}, s_i) \in \rho(\alpha)$ *für alle* $1 \leq i \leq k\}$
8. $\rho(F?) = \{(s,s) \mid s \in \tau(F)\}$

In Abbildung 3.9 sind einige der gängigen Programmkonstrukte und ihre Definitionen in dynamischer Logik zusammengestellt. Daraus wird klar wie PDL_{while}in PDL_{reg}eingebettet werden kann.

Um etwas Vertrautheit mit der Notation für Programme in der dynamischen Logik zu erwerben betrachten wir die Definition der while-Schleife etwas genauer. Zunächst bemerken wir

$$\rho(A?;\alpha) = \{(s,t) \mid s \in \tau(A) \text{ und } (s,t) \in \rho(\alpha)\}$$

Mit dieser Information ausgestattet rechnen wir die Interpretation für die angegebene Definition der while-Schleife aus:

$$\begin{aligned}
&\rho((A?;\alpha)^*;\neg A?)\\
&\quad = \{(s,t) \in \rho((A?;\alpha)^*) \mid t \in \tau(\neg A)\}\\
&\quad = \{(s,t) \mid \text{ es gibt } 0 \leq k \text{ und } s_0, \ldots, s_k \text{ mit}\\
&\qquad s_0 = s\\
&\qquad s_k = t
\end{aligned}$$

$(s_{i-1}, s_i) \in \rho(\alpha)$ für alle i mit $1 \leq i \leq k$
$s_i \in \tau(A)$ für alle i mit $0 \leq i < k$ und
$t \notin \tau(A)\}$

Man beachte, daß für $s \notin \tau(A)$ auch $(s, s) \in \rho((A?; \alpha)^*; \neg A?)$ gilt. Das ist der Fall $k = 0$.

Ist s ein Zustand in einer Struktur $\mathcal{A}$ und α ein erlaubtes Programm, dann wird die Aussage, daß α im Zustand s terminiert in der Terminologie der dynamischen Logik durch die Aussage, es gibt ein $t \in W$ mit $(s, t) \in \rho(\alpha)$ ausgedrückt.

Definition 88 *Sei $\mathcal{A} = (W, \tau, \rho)$ eine Struktur der dynamischen Aussagenlogik $w \in W$ und $F \in Fdal$. Wir benutzen die folgende Terminologie*

w erfüllt F	$w \models F$	*für*	$w \in \tau(F)$.
F ist $\mathcal{A}$-gültig	$\mathcal{A} \models F$	*für*	*$w \in \tau(F)$ für alle $w \in W$.*
F ist allgemeingültig	$\models F$	*für*	*F ist $\mathcal{A}$-gültig für alle $\mathcal{A}$*

Definition 89 (Das logische System PDL_{min})
Das logische System PDL_{min}besteht aus den folgenden Axiomen und Regeln

Axiome

Alle Tautologien der Aussagenlogik			*(A1)*
$<\alpha> (F \vee G)$	$\leftrightarrow$	$<\alpha> F \vee <\alpha> G$	*(A2)*
$<\alpha;\beta> F$	$\leftrightarrow$	$<\alpha><\beta> F$	*(A3)*
$<\alpha \cup \beta> F$	$\leftrightarrow$	$<\alpha> F \vee <\beta> F$	*(A4)*
$<\alpha^*> F$	$\leftrightarrow$	$F \vee <\alpha><\alpha^*> F$	*(A5)*
$<A?> F$	$\leftrightarrow$	$A \wedge F$	*(A6)*
$[\alpha^*](F \to [\alpha]F)$	$\to$	$(F \to [\alpha^*]F)$	*(A7)*
$[\alpha](F \to G)$	$\to$	$[\alpha]F \to [\alpha]G$	*(A8)*

Regeln

$$\frac{F,\ F \to G}{G} \qquad \textit{(MP)}$$

$$\frac{F}{[\alpha]F} \qquad \textit{(G)}$$

Mit $\vdash_{DAL}$ bezeichnen wir die Ableitbarkeitsrelation im System PDL_{min}.

Lemma 76 *In DAL gelten die folgenden abgeleiteten Regeln:*

$$\frac{F \to [\alpha]F}{F \to [\alpha^*]F} \qquad \textit{(I)}$$

$$\frac{F \to G}{[\alpha]F \to [\alpha]G} \qquad \textit{(D)}$$

$$\frac{F \to G,\ G \to H}{F \to H} \qquad \textit{(T)}$$

Beweis:
(I) Die Axiome und Regeln aus Definition 89 sind Schemata, anwendbar für jede Instanziierung der darin vorkommenden Formeln F, G und Programme α, β. Aus $F \to [\alpha]F$ kann man mit einer Anwendung der Regel (G) mit α^* anstelle von α in einem Schritt $[\alpha^*](F \to [\alpha]F)$ herleiten. Mit (A7) und (MP) folgt daraus die gewünschte Konklusion $F \to [\alpha^*]F$.
(D) Aus $F \to G$ folgt mit (G) $[\alpha](F \to G)$, woraus mit Axiom (A8) und modus ponens (MP) die Konklusion $[\alpha]F \to [\alpha]G$ folgt.
(T) Reine Aussagenlogik.

$\blacksquare$

Lemma 77 *Die folgenden Formeln sind aus DAL und der Definition*

$$[\alpha]F \leftrightarrow \neg <\alpha> \neg F$$

herleitbar

$$\begin{array}{lll}
\mathit{1} & [\alpha](F \wedge G) & \leftrightarrow & [\alpha]F \wedge [\alpha]G \\
\mathit{2} & [\alpha;\beta]F & \leftrightarrow & [\alpha][\beta]F \\
\mathit{3} & [\alpha \cup \beta]F & \leftrightarrow & [\alpha]F \wedge [\beta]F \\
\mathit{4} & [\alpha^*]F & \leftrightarrow & F \wedge [\alpha][\alpha^*]F
\end{array}$$

Beweis: Die vier Formeln folgen aus der Definition und jeweils aus den Axiomen (A2), (A3), (A4) und (A5) mit Aussagenlogik.

$\blacksquare$

Satz 78 (Korrektheitssatz für DAL)
Für jede Formel F aus $Fml_{PDL_{reg}}$ gilt:

$$aus\ \vdash_{DAL} F\ folgt \models F$$

Beweis: Einfaches Nachrechnen.

$\blacksquare$

Satz 79 (Vollständigkeitssatz für DAL)
Für jede Formel F aus $Fml_{PDL_{reg}}$ gilt:

$$aus\ \models F\ folgt \vdash_{DAL} F$$

Beweis: Siehe z.B. [Harel, 1984], pp 512 - 515.

Definition 90 (Fischer-Ladner Abschluß)
Ist F eine Formel in $Fml_{PDL_{reg}}$ dann heißt die kleinste Formelmenge S, welche die folgenden Abschlußeigenschaften besitzt, der Fischer-Ladner Abschluß von F, bezeichnet mit $FL(F)$:

$$\begin{array}{lrcl}
1 & F \in S & & \\
2 & \neg G \in S & \Rightarrow & G \in S \\
3 & (G_1 \vee G_2) \in S & \Rightarrow & G_1 \in S \textit{ and } G_2 \in S \\
4 & <\alpha> G \in S & \Rightarrow & G \in S \\
5 & <\alpha;\beta> G \in S & \Rightarrow & <\alpha><\beta> G \in S \\
6 & <\alpha \cup \beta> G \in S & \Rightarrow & <\alpha> G \in S \textit{ und } <\beta> G \in S \\
7 & <\alpha^*> G \in S & \Rightarrow & <\alpha><\alpha^*> G \in S \\
8 & <G_1?> G_2 \in S & \Rightarrow & G_1 \in S \textit{ und } G_2 \in S
\end{array}$$

Definition 91
Ist $\mathcal{A} = (W, \tau, \rho)$ *eine* $Fml_{PDL_{reg}}$*-Struktur und* S *eine Menge von* $Fml_{PDL_{reg}}$*-Formeln, dann definieren wir die Äquivalenzrelation* $\sim_S$ *auf der Menge* W *aller Zustände durch*

$$w_1 \sim_S w_2 \textit{ gdw } w_1 \models F \Leftrightarrow w_2 \models F \textit{ für alle } F \in S$$

Vergleiche Definition 57.

Definition 92
Die Qutientenstruktur $\mathcal{A}_S = (W_S, \tau_S, \rho_S)$ *zu einer Struktur* $\mathcal{A}$ *modulo der Äquivalenzrelation* $\sim_S$ *wird für atomare Formeln* p *und atomare Programme* $\pi \in \Pi_0$ *wie üblich definiert:*

$$\begin{array}{lcll}
W_S & = & \{[w] \mid w \in W\} & [w] \textit{ ist die Äquivalenzklasse von } w \\
\tau_S(p) & = & \{[w] \mid w \in \tau(p)\} & \textit{falls } p \in S \\
\tau_S(p) & = & \textit{beliebig} & \textit{sonst}
\end{array}$$

Die Definition der Übergangsrelationen $\rho_S(\pi)$ *ist etwas komplizierter:*

$$\begin{array}{lll}
([w_1],[w_2]) \in \rho_S(\pi) & \textit{gdw} & \textit{für alle } <\pi> B \in S \textit{ gilt} \\
 & & w_1 \models \neg <\pi> B \Rightarrow w_2 \models \neg B
\end{array}$$

Wir haben um die Anzahl der Fallunterscheidungen zu verringern nur die Modaloperatoren $<\alpha>$ in den Formeln in S zugelassen. Die Operatoren $[\alpha]$ können ja durch $\neg <\alpha> \neg$ ersetzt werden. Führt man in der Definition von $\rho_S(\pi)$ den Boxoperator wieder ein, so erhält man die äquivalente Bedingung:

$$\begin{array}{lll}
([w_1],[w_2]) \in \rho_S(\pi) & \textit{gdw} & \text{für alle } [\pi]B \in S \text{ gilt} \\
 & & w_1 \models [\pi]B \Rightarrow w_2 \models B
\end{array}$$

Man vergleiche diese Definition mit der in Aufgabe 3.4.1 beschriebenen modalen Zugänglichkeitsrelation R.

Lemma 80
Sei F eine $Fml_{PDL_{reg}}$-Formel, $S = FL(F)$ ihr Fischer-Ladner Abschluß, $\mathcal{A} = (W, \tau, \rho)$ eine $Fml_{PDL_{reg}}$-Struktur und $\mathcal{A}_S = (W_S, \tau_S, \rho_S)$ ihre Quotientenstruktur modulo $\sim_S$, dann gilt für alle $G \in S$, $\alpha \in \Pi$ und $w_1, w_2 \in W$

1. *Aus $([w_1], [w_2]) \in \rho_S(\alpha)$ folgt für alle $< \alpha > B \in S$ gilt* $w_1 \models \neg < \alpha > B \Rightarrow w_2 \models \neg B$
2. *aus $(w_1, w_2) \in \rho(\alpha)$ folgt $([w_1], [w_2]) \in \rho_S(\alpha)$*
3. *$w \models G$ gdw $[w] \models G$*

Beweis: Der Beweis der drei Behauptungen erfolgt durch simultane Rekursion über die Komplexität von α bzw. von G. Für atomare Programme und Formeln reduzieren sich alle drei Aussagen auf die entsprechenden Definitionen.

zu Teil 1 Wir führen den Induktionsschritt von $< \alpha; \beta >$ auf $< \alpha >$ und $< \beta >$ als ein Beispiel vor. Gelte also $([w_1], [w_2]) \in \rho_S(\alpha; \beta)$. Nach Definition der Hintereinanderausführung gibt es also ein u mit

$$\begin{array}{lcl} ([w_1], [u]) & \in & \rho_S(\alpha) \text{ und} \\ ([u], [w_2]) & \in & \rho_S(\beta) \end{array}$$

Nach Induktionsvoraussetzung gilt für alle $< \alpha > B \in S$ und alle $< \beta > C \in S$

$$w_1 \models \neg < \alpha > B \Rightarrow u \models \neg B \quad (1)$$
$$u \models \neg < \beta > C \Rightarrow w_2 \models \neg C \quad (2)$$

Zu zeigen ist, daß für alle $< \alpha; \beta > F \in S$ aus $w_1 \models \neg < \alpha; \beta > F$ folgt $w_2 \models \neg F$. Gelte also $w_1 \models \neg < \alpha; \beta > F$. Nach Definition des Fischer-Ladner-Abschlusses gilt auch $< \alpha >< \beta > F \in S$ und damit auch $< \beta > F \in S$. Aus der Annahme folgt unmittelbar

$$w_1 \models \neg < \alpha >< \beta > F$$

Aus (1) mit $B =< \beta > F$ folgt:

$$u \models \neg < \beta > F$$

Woraus mit (2) schließlich wie gewünscht

$$w_2 \models \neg F$$

folgt.

Als nächstes betrachten wir den Induktionsschritt für $< \alpha^* >$. Sei also ein Paar $([w_1], [w_2])$ aus $\rho_S(< \alpha^* >)$ gegeben. Nach Definition von $\rho_S(< \alpha^* >)$ gibt es Zustände $u_0, \ldots, u_k$ mit $[w_1] = [u_0]$, $[w_2] = [u_k]$ und für alle $0 \leq i < k$ gilt $([u_i], [u_{i+1}]) \in \rho_S(< \alpha >)$. Nach der Induktionsvoraussetzung für Teil 1 gilt für alle $< \alpha > C \in S$

$$u_i \models \neg < \alpha > C \Rightarrow u_{i+1} \models \neg C$$

Wir müssen für alle Formeln $< \alpha^* > B \in S$ zeigen

$$w_1 \models \neg < \alpha^* > B \Rightarrow w_2 \models \neg B$$

Als erstens bemerken wir daß die Formel

$$< \alpha^* > B \leftrightarrow B \vee < \alpha >< \alpha^* > B$$

eine Tautologie der dynamischen Aussagenlogik ist, also auch

$$\neg < \alpha^* > B \leftrightarrow \neg B \wedge \neg < \alpha >< \alpha^* > B$$

Ausgehen von $u_0 \models \neg < \alpha^* > B$ zeigen durch Induktion nach i für $0 \leq i \leq k$, daß $u_i \models \neg < \alpha^* > B$ gilt. Wir gerade bemerkt folgt aus $u_i \models \neg < \alpha^* > B$ auch $u_i \models \neg < \alpha >< \alpha^* > B$. Nach Definition des Fischer-Ladner-Abschlusses liegt $< \alpha >< \alpha^* > B$ in S. Daher ist die Induktionvoraussetzung mit $C =< \alpha^* > B$ anwendbar und liefert $u_{i+1} \models \neg < \alpha^* > B$. Aus $u_k \models \neg < \alpha^* > B$ folgt dann wegen der oben genannten Tautologie direkt $u_k \models \neg B$ und wegen $[u_k] = [w_2]$ auch $w_2 \models \neg B$.

Abschließend betrachten wir den Fall $([w_1], [w_2]) \in \rho_S(< F? >)$. Daraus folgt $[w_1] = [w_2$ und $[w_1] \models F$. Für $< F? > B \in S$ nehmen wir $w_1 \models \neg < F? > B$ an mit dem Ziel $w_2 \models \neg B$ zu zeigen. Zunächst können wir schließen auf $w_1 \models \neg F \vee \neg B$. Aus $[w_2] \models F$ folgt aber mit Teil 3 $w_2 \models F$, woraus dann, wie gewünscht, $w_2 \models \neg B$ folgt.

zu Teil 2 Sei $(w_1, w_2) \in \rho(\alpha)$. Um $([w_1], [w_2]) \in \rho_S(\alpha)$ zu zeigen, betrachten wir gemäß Definition von ρ_S eine Formel $\neg < \alpha > B$ in S mit $w_1 \models \neg < \alpha > B$. Das heißt mit $w_1 \models [\alpha]\neg B$. Somit gilt, wie verlangt, $w_2 \models \neg B$.

zu Teil 3 Die Induktionsschritte, welche die aussagenlogischen Verknüpfungen in G betreffen sind einfach. Wir wollen den Schritt von B nach $< \alpha > B$ vorführen.

Gilt $w_1 \models < \alpha > B$, so gibt es einen Zustand w_2 mit $(w_1, w_2) \in \rho(\alpha)$ und $w_2 \models B$. Nach Induktionsvoraussetzung gilt auch $[w_2] \models B$ und nach Teil 2 auch $([w_1], [w_2]) \in \rho_S(\alpha)$ und damit auch $[w_1] \models < \alpha > B$. Nehmen wir jetzt umgekehrt an, daß $[w_1] \models < \alpha > B$ gilt, so gibt es ein $[w_2]$ mit $([w_1], [w_2]) \in \rho_S(\alpha)$ und $[w_2] \models B$ und damit nach Induktionsvoraussetzung auch $w_2 \models B$. Würde nun $w_1 \models \neg < \alpha > B$ gelten, so würde nach Teil 1 für $\rho_s(\alpha)$ der Widerspruch $w_2 \models \neg B$ folgen.

∎

Satz 81 (Entscheidbarkeit)
Das Erfüllbarkeitsproblem für PDL_{reg}ist entscheidbar.

Damit ist auch das Erfüllbarkeitsproblem für PDL_{while}ist entscheidbar.

Beweis: Sei F eine Formel in $Fml_{PDL_{reg}}$. Ihr Fischer-Ladner Abschluß $FL(F)$ ist eine endliche Formelmenge. Sei n die Kardinalität von $FL(F)$. Wir behaupten:

> Wenn F erfüllbar ist, dann besitzt F schon ein Modell mit höchstens 2^n Welten. Daraus folgt unmittelbar die Entscheidbarkeit. Sei $\mathcal{A}$ eine erfüllende Struktur für F, d.h. es gilt $(\mathcal{A}, w) \models F$ fü eine Welt w von $\mathcal{A}$. Die Quotientenstruktur $\mathcal{A}_S$ für $S = FL(F)$ (siehe Definition 92) besitzt höchstens 2^n Welten und nach Lemma 80 is F auch in $\mathcal{A}_S$.
>
> Siehe auch [Harel, 1984], Theorem 2.12 auf S. 515.

∎

3.8.4 Dynamische Quantorenlogik

Die dynamische Quantorenlogik, QDL, ist eine Erweiterung der Prädikatenlogik erster Stufe. Insbesondere ist zur Festlegung einer Instanz der dynamischen Quantorenlogik die Angabe eines Vokabulars V erforderlich, das die erlaubten Funktions- und Konstantenzeichen enthält. Das Gleichheitszeichen $=$ soll stets in V vorhanden sein und mit Var werde die Menge aller Individuenvariablen bezeichnet.

Definition 93 (Formeln und Programme von QDL_{while})
Die Menge der Formeln der dynamischen Quantorenlogik $Fml_{QDLwhile}$ und der erlaubten Programme Π_{while} wird simultan induktiv definiert durch

1. *die Konstanten* wahr *und* falsch *sind in $Fml_{QDLwhile}$,*
2. *jede atomare Formel ist in $Fml_{QDLwhile}$,*
3. *sind $F_1, F_2 \in Fml_{QDLwhile}$ und x eine Variable in Var, dann sind auch $\neg F_1$, $(F_1 \vee F_2)$ und $\exists x F_1$ Formeln in $Fml_{QDLwhile}$,*
4. *für $F \in Fml_{QDLwhile}$ und $\alpha \in \Pi_{while}$ gilt $< \alpha > F \in Fml_{QDLwhile}$,*
5. *die Menge der Programme Π_{while} für QDL_{while} erhält man aus Definition 83, indem man für die Menge der atomaren Programme Π_0 die Zuweisungs einsetzt, i.e.*

$$\Pi_0 = \{x := t \mid \text{ für } x \in \ Var \text{ und } t \text{ ein Term}\}$$

Wir benutzen stillschweigend auch die definierbaren logischen Operatoren $\wedge$, $\rightarrow$, $\leftrightarrow$, $\forall x$ und $[\alpha]$.
Man beachte, daß in prädikatenlogischen Formeln und in Programmen dieselben Individuenvariablen benutzt werden. Häufig werden in Formulierungen der dynamischen Quantorenlogik Testformeln in $while$-Schleifen und if-Konstrukten auf quantorenfreien, prädikatenlogische Formeln beschränkt.
Die Semantik von Formeln aus $Fml_{QDLwhile}$ wird durch eine übliche Struktur $\mathcal{M}$ der Prädikatenlogik erster Stufe festgelegt, mit einem Universum $U_{\mathcal{M}}$ und Interpretationen $f_{\mathcal{M}}$ und $R_{\mathcal{M}}$ der

Funktions- und Relationszeichen. Die Menge W der Zustände ist die Menge aller Belegungen von Individuenvariablen, d.h. $W = \{w \mid w : Var \rightarrow U_{\mathcal{M}}\}$. Ausgehend von $\mathcal{M}$ können die Abbildungen τ und ρ wie in Definition 85 bestimmt werden, wobei an die Stelle aussagenlogischer Variablen jetzt atomare Formeln treten und die Stelle atomarer Programme jetzt durch Zuweisungen $(x := t)$ eingenommen wird. Die einzigen nachzutragenden Klauseln der Semantikdefinition sind also

Definition 94 (Semantik von QDL_{while})

1. *$w \in \tau(R(t_1, \ldots, t_k))$ gdw die atomare Formel $R(t_1, \ldots, t_k)$ ist wahr in der Struktur $\mathcal{M}$ unter der Belegung der Variablen w.*
2. *$\rho(x := t) = \{(w_1, w_2) \mid w_2(y) = w_1(y)$ für $x \neq y$ und $w_2(x) = I(\mathcal{M}, w_1)(t)\}$*
   wobei $I(\mathcal{M}, w_1)(t)$ die Interpretation des Terms t in $\mathcal{M}$ unter der Belegung w_1 ist.

Wir schreiben auch $(\mathcal{M}, w) \models F$, gelesen als „die Formel F ist wahr im Zustand w in $\mathcal{M}$“, anstelle von $w \in \tau(F)$.

Definition 95 *Wir sagen F ist* **gültig** *in $\mathcal{M}$, falls für jeden Zustand w gilt $(\mathcal{M}, w) \models F$.*
F heißt **allgemeingültig**, *falls $\mathcal{M} \models F$ für jede Struktur $\mathcal{M}$ gilt.*

Satz 82 (Uninterpretierter Fall)
Die Menge aller allgemeingültigen Formeln in $Fml_{QDLwhile}$ ist nicht rekursiv aufzählbar.

Beweis: siehe [Harel, 1984] Theorem 3.7, p.551.
Nach diesem Satz ist eine vollständige und korrekte Axiomatisierung der dynamischen Quantorenlogik nicht möglich. Erweitert man den Begriff eines Axiomensystems und läßt auch Regeln mit unendlich vielen Prämissen zu, was zumindest theoretisch durchaus Sinn macht, so kann eine vollständige und korrekte infinitäre Axiomatisierung gegeben werden, siehe [Harel, 1984] pp. 559-560.

Neben dem eben angesprochenen uninterpretierten Fall ist mit Rücksicht auf praktische Belange der interpretierte Fall von größerer Bedeutung. Hierbei werden die Funktions- und Relationszeichen des Vokabulars V in einer fest gewählten Struktur $\mathcal{M}$ interpretiert und gefragt wird nach den in $\mathcal{M}$ gültigen Formeln. Die Struktur $\mathcal{M}$ wird üblicherweise die Datenstrukturen der betrachteten Programme enthalten, etwa die natürlichen Zahlen, Listen von Basisobjekten und ähnliches. In der Regel wird dabei schon die Menge der in $\mathcal{M}$ gültigen Formeln der Prädikatenlogik erster Stufe schon nicht mehr rekursiv aufzählbar sein, so daß auch im interpretierten Fall kein vollständiges und korrektes Axiomensystem für die dynamische Prädikatenlogik zu erwarten ist. Interessanterweise gilt jedoch ein relativer Vollständigkeitssatz, zumindest für Strukturen $\mathcal{M}$, die einer gewissen Reichhaltigkeitsforderung genügen. Dabei werden alle in $\mathcal{M}$ gültigen Formeln erster Stufe zu den Axiomen hinzugenommen.

Definition 96 *Eine Struktur $\mathcal{M}$ heißt eine* **arithmetische Struktur***, wenn sie die natürliche Zahlen in definierbarer Weise enthält, d.h. es gibt Formeln $U(x)$,$A(x,y,z)$,$M(x,y,z)$ in der Prädikatenlogik erster Stufe, so daß die Struktur (U,a,m) isomorph zur Struktur $(I\!N,+,\times)$ der natürlichen Zahlen ist, wobei*

$$\begin{array}{lcl} U & = & \{g \in U_{\mathcal{M}} \mid \mathcal{M} \models U(g)\} \\ a(g,h)=k & gdw & \mathcal{M} \models A(g,h,k) \\ m(g,h)=k & gdw & \mathcal{M} \models M(g,h,k) \end{array}$$

Definition 97 (Das logische System $QDL_{\mathcal{M}}$)
Sei $\mathcal{M}$ eine Struktur.
Das logische System $QDL_{\mathcal{M}}$ besteht aus den folgenden Axiomen und Regeln

Axiome

Alle Tautologien der Aussagenlogik	*(A1)*
Alle in $\mathcal{M}$ gültigen Formeln des PK1	*(A2)*
$< x := t > F \quad \leftrightarrow \quad F(t/x)$	*(A3)*

Regeln

$$\frac{F,\ F \rightarrow G}{G} \qquad \text{(MP)}$$

$$\frac{F}{[\alpha]F} \qquad \text{(G)}$$

Wir schreiben $\vdash_{\mathcal{M}} F$, wenn F in dem System $QDL_{\mathcal{M}}$ herleitbar ist.

Satz 83 (Relative Vollständigkeit)
Sei $\mathcal{M}$ eine arithmetische Struktur. Dann gilt für jede Formel $F \in Fml_{QDLwhile}$:

$$\mathcal{M} \models F \textit{ genau dann, wenn } \vdash_{\mathcal{M}} F$$

Beweis: siehe [Harel, 1984], Theorem 3.19, p.565.

3.8.5 Ein Sequenzenkalkül für QDL_{while}

Eine Sequenz $\Gamma \ \vdash \ \Delta$ drückt intuitiv die Tatsache aus, daß $\bigvee \Delta$ eine logische Folgerung aus $\bigwedge \Gamma$ ist. Eine Sequenz $\Gamma \ \vdash \ \Delta$ ist also allgemeingültig, wenn die Formel $\bigwedge \Gamma \rightarrow \bigvee \Delta$ allgemeingültig ist.

In Abbildung 3.10 auf Seite 234 sind die Regeln für ein Sequenzenkalkül für eine Logik erster Stufe mit Peano Arithmetik zusammengestellt. Wir betrachten im folgenden einzeln die Regeln für Formeln der Dynamischen Logik.

Kompositionsregel:

$$\frac{\Gamma \ \vdash \ [\alpha_1][\alpha_2; \ldots; \alpha_n]\, \phi, \Delta}{\Gamma \ \vdash \ [\alpha_1; \alpha_2; \ldots; \alpha_n]\, \phi, \Delta}$$

Zuweisungsregel:

$$\frac{\Gamma\{X \leftarrow X'\}, X \doteq t\{X \leftarrow X'\} \;\vdash\; \phi, \Delta\{X \leftarrow X'\}}{\Gamma \;\vdash\; [X := t]\,\phi, \Delta}$$

wobei X' eine neue Variable ist.

Wie schon früher erwähnt gibt es in der Dynamischen Logik keine syntaktische Unterscheidung zwischen logischen Variablen und Programmvariablen. Die Zuweisungsregel ist in diesem Zusammenhang von entscheidender Bedeutung, da durch ihre Anwendung eine Programmvariable X Anlaß gibt zur Einführung einer neuen *logischen* Variable. Betrachten wir ein kleines Beispiel. Die Beweisverpflichtung $even(X) \;\vdash\; [X := X + 2]\,even(X)$ sei zu bearbeiten. Die Anwendung der Zuweisungsregel führt zu $even(X'), X \doteq X'+2 \;\vdash\; even(X)$. Was offensichtlich eine wahre Aussage über die natürlichen Zahlen ist.

Axiome:

$$\frac{}{\phi, \Gamma \ \vdash \ \phi, \Delta} \qquad \frac{}{\text{false}, \Gamma \ \vdash \ \Delta} \qquad \frac{}{\Gamma \ \vdash \ \text{true}, \Delta}$$

Aussagenlogische Regeln:

$$\frac{\Gamma \ \vdash \ \phi, \Delta}{\neg\phi, \Gamma \ \vdash \ \Delta} \qquad \frac{\phi, \Gamma \ \vdash \ \Delta}{\Gamma \ \vdash \ \neg\phi, \Delta}$$

$$\frac{\phi, \psi, \Gamma \ \vdash \ \Delta}{\phi \wedge \psi, \Gamma \ \vdash \ \Delta} \qquad \frac{\Gamma \ \vdash \ \phi, \Delta \qquad \Gamma \ \vdash \ \psi, \Delta}{\Gamma \ \vdash \ \phi \wedge \psi, \Delta}$$

Regeln für Quantoren:

$$\frac{\phi\{X \leftarrow t\}, \forall X\, \phi, \Gamma \ \vdash \ \Delta}{\forall X\, \phi, \Gamma \ \vdash \ \Delta} \qquad \frac{\Gamma \ \vdash \ \phi\{X \leftarrow t\}, \exists X\, \phi, \Delta}{\Gamma \ \vdash \ \exists X\, \phi, \Delta} \quad t \text{ Term}$$

$$\frac{\phi\{X \leftarrow Y\}, \Gamma \ \vdash \ \Delta}{\exists X\, \phi, \Gamma \ \vdash \ \Delta} \qquad \frac{\Gamma \ \vdash \ \phi\{X \leftarrow Y\}, \Delta}{\Gamma \ \vdash \ \forall X\, \phi, \Delta} \qquad Y\,\textbf{neu}$$

Schnittregel:

$$\frac{\Gamma \ \vdash \ \phi, \Delta \qquad \phi, \Gamma \ \vdash \ \Delta}{\Gamma \ \vdash \ \Delta}$$

Strukturelle Regeln: Umordnen, Abschwächen und Verstärken.

Gleichheitsregeln:

$$\frac{}{\vdash \ t \doteq t} \qquad \frac{s \doteq t, \Gamma_s^t \ \vdash \ \Delta_s^t}{s \doteq t, \Gamma \ \vdash \ \Delta} \qquad \frac{t \doteq s, \Gamma_s^t \ \vdash \ \Delta_s^t}{t \doteq s, \Gamma \ \vdash \ \Delta}$$

Regel zur Peano Arithmetik:

$$\frac{}{\Gamma \ \vdash \ \Delta}$$

falls kein Programm in Γ, Δ auftritt und $\bigwedge \Gamma \models_{\text{Peano}} \bigvee \Delta$ gilt

Abbildung 3.10: Regeln für einen Sequenzenkalkül

Verzweigungsregel:

$$\frac{\Gamma, B \ \vdash \ [\alpha]\,\phi, \Delta \qquad\qquad \Gamma, \neg B \ \vdash \ [\beta]\,\phi, \Delta}{\Gamma \ \vdash \ [\underline{if} \ \ B \ \underline{then} \ \ \alpha \ \underline{else} \ \ \beta \ \underline{end}\,]\,\phi, \Delta}$$

Schleifenregel:

$$\frac{\Gamma \ \vdash \ Inv \qquad\qquad Inv, B \ \vdash \ [\alpha]\,Inv \qquad\qquad Inv, \neg B \ \vdash \ \phi, \Delta}{\Gamma \ \vdash \ [\ \underline{while} \ \ B \ \underline{do} \ \ \alpha \ \underline{end}\,]\,\phi, \Delta}$$

Inv heißt eine Schleifeninvariante.

$$\begin{array}{c}
I := 0; \\
Y := 0; \\
\underline{while} \ \ I < X \ \underline{do} \\
Y := Y + 2 * I + 1; \\
I := I + 1 \\
\underline{end}
\end{array}$$

Abbildung 3.11: Das Programm α_{sq}

Als ein Beispiel für das Arbeiten mit dem Sequenzenkalkül betrachten wir das Programm α_{sq} aus Abbildung 3.11 und wollen zeigen, daß $\vdash [\alpha_{\text{square}}]\,Y \doteq X * X$ gilt. Nach den ersten vier Beweisschritten ist der folgende Ableitungsbaum entstanden.

$$\begin{array}{rcl}
\multicolumn{3}{c}{I \doteq 0,\ Y \doteq 0 \ \vdash \ Inv \qquad Inv,\ B \ \vdash \ [\alpha]\,Inv \qquad Inv,\ \neg B \ \vdash \ Y \doteq X * X} \\
I \doteq 0,\ Y \doteq 0 & \vdash & [\underline{while} \ \ B \ \underline{do} \ \ \alpha \ \underline{end}\,]\,Y \doteq X * X \\
I \doteq 0 & \vdash & [Y := 0]\,[\underline{while} \ \ B \ \underline{do} \ \ \alpha \ \underline{end}\,]\,Y \doteq X * X \\
 & \vdash & [I := 0]\,[Y := 0]\,[\underline{while} \ \ B \ \underline{do} \ \ \alpha \ \underline{end}\,]\,Y \doteq X * X \\
 & \vdash & [I := 0; \ \ Y := 0; \ \ \underline{while} \ \ B \ \underline{do} \ \ \alpha \ \underline{end}\,]\,Y \doteq X * X \\
 & \vdash & [\alpha_{\text{square}}]\,Y \doteq X * X
\end{array}$$

B: $\quad I < X$
α: $\quad Y := Y + 2 * I + 1; \ \ I := I + 1$
Inv: $\quad I \leq X \ \wedge \ Y \doteq I * I$

Bei der Entwicklung eines solchen Beweisbaums hat es sich eingebürgert von unten nach oben zu arbeiten. Die erste Zeile (von unten gelesen) ist also das Beweisziel $\vdash [\alpha_{\text{square}}]\,Y \doteq X * X$ und die darüberstehenden Zeilen entstehen durch Regelanwendung, *von unten nach oben*. Es sind jetzt drei Beweisziele offen.

Inv und negierte Schleifenbedingung impliziert Nachbedingung

$$\begin{array}{rcl} I \leq X,\ Y \doteq I * I,\ \neg(I < X),\ I \doteq X & \vdash & Y \doteq I * I \\ I \leq X,\ Y \doteq I * I,\ \neg(I < X)\ I \doteq X & \vdash & Y \doteq X * X \\ I \leq X,\ Y \doteq I * I,\ \neg(I < X) & \vdash & Y \doteq X * X \\ Inv \wedge \neg B & \vdash & Q \end{array}$$

Inv bleibt unter α erhalten

$$\begin{array}{c} I' < X,\ I \doteq I' + 1 \ \vdash \ I \leq X \\ Y' \doteq I' * I',\ Y \doteq Y' + 2 * I' + 1,\ I \doteq I' + 1 \ \vdash \ Y \doteq I * I \\ I' \leq X,\ Y' \doteq I' * I',\ I' < X,\ Y \doteq Y' + 2 * I' + 1,\ I \doteq I' + 1 \quad \vdash \quad Inv \\ I \leq X,\ Y' \doteq I * I,\ I < X,\ Y \doteq Y' + 2 * I + 1 \quad \vdash \quad [I := I + 1]\, Inv \\ I \leq X,\ Y \doteq I * I,\ I < X \quad \vdash \quad [Y := Y + 2 * I + 1]\, [I := I + 1]\, Inv \\ I \leq X,\ Y \doteq I * I,\ I < X \quad \vdash \quad [Y := Y + 2 * I + 1;\ I := I + 1]\, Inv \\ Inv,\ B \quad \vdash \quad [\alpha]\, Inv \end{array}$$

Vorbedingung impliziert Inv

$$\begin{array}{l} I \doteq 0,\ Y \doteq 0 \ \vdash \ 0 \leq X \qquad\qquad I \doteq 0,\ Y \doteq 0 \ \vdash \ Y \doteq 0 * 0 \\ I \doteq 0,\ Y \doteq 0 \ \vdash \ 0 \leq X \ \wedge \ Y \doteq 0 * 0 \\ I \doteq 0,\ Y \doteq 0 \ \vdash \ I \leq X \ \wedge \ Y \doteq I * I \end{array}$$

Der erste Teilbeweisbaum endet in einem Axiom, auf beiden Seite des Sequenzenzeichens $\vdash$ kommt die Formel $Y \doteq I * I$ vor. Der zweite Beweisbaum endet ist noch nicht an aeinem Axiom angekommen. Man sieht aber leicht, daß aus $I' < X,\ I \doteq I' + 1$ mit Hilfe der Axiome der Peano-Arithmetik $I \leq X$ gefolgert werden kann und wir sind auch hier erfolgreich. Im dritten Beweisteilbaum endet der rechte Ast wieder in einem Axiom, aber der linke Ast bleibt offen. Und das mit gutem Grund, denn das Programm α_{sq} ist nicht korrekt ohne die zusätzliche Annahme $X \geq 0$.

Nachdem wir einige Beispiele gesehen haben sind wir bereit für die allgemeine Definition:

Definition 98 *Eine* Beweisbaum *im Sequenzenkalkül ist ein Baum dessen Knoten mit Sequenzen $\Delta \vdash \Gamma$ markiert sind, so daß für jeden innern Knoten N des Baums mit den Nachfolgerknoten $N_1, \ldots, N_k$ es eine Kalkülregel der Form*

$$\frac{\Gamma_1 \ \vdash \ \Delta_1 \qquad \ldots \qquad \Gamma_k \ \vdash \ \Delta_k}{\Gamma \ \vdash \ \Delta}$$

gibt, so daß N mit der Sequenz $\Gamma \vdash \Delta$*markiert ist und die Nachfolgerknoten* N_i *mit der Sequenz* $\Gamma_i \vdash \Delta_i$

Ein Beweisbaum $\mathcal{B}$ *heißt* geschlossen, *wenn alle seine Blätter mit Axiomen markiert sind. Wir nennen* $\mathcal{B}$ *einen Beweis für die Sequenz* $\Gamma \vdash \Delta$*, wenn die Wurzel von* $\mathcal{B}$ *mit dieser Sequenz markiert ist.*

Definition 99 *Ein Sequenzenkalkül heißt* korrekt *wenn für jeden geschlossenen Beweisbaum die Wurzelmarkierung allgemeingültig ist.*

Eine Kalkülregel

$$\frac{\Gamma_1 \vdash \Delta_1 \qquad \ldots \qquad \Gamma_k \vdash \Delta_k}{\Gamma \vdash \Delta}$$

heißt korrekt, wenn aus der Allgemeingültigkeit von $\Gamma_1 \vdash \Delta_1$ *und ... und* $\Gamma_k \vdash \Delta_k$ *die Allgemeingültigkeit von* $\Gamma \vdash \Delta$ *folgt.*

Lemma 84 *Sind alle Beweisregeln eines Kalküls korrekt, so ist der Kalkül selbst korrekt.*

Beweis: Wie üblich.

■

Lemma 85 *Die Regeln des Sequenzenkalküls* QDL_{while} *sind korrekt. korrekt.*

Beweis: Wir beweisen als ein beispiel die Korrektheit der Schleifenregel.

Vorausgesetzt ist also die Allgemeingültigkeit der drei Implikationen $\bigwedge \Gamma \rightarrow Inv$, $Inv \wedge B \rightarrow [\alpha]\, Inv$ und $Inv \wedge \neg B \rightarrow \phi \vee \bigvee \Delta$.

Unser Ziel ist $\Gamma \rightarrow [\underline{while}\ B\ \underline{do}\ \ \alpha\ \underline{end}\ ]\, \phi \vee \bigvee \Delta$ zu zeigen.

Wir beginnen mit einem Zustand $(\mathcal{M}, w)$ mit $(\mathcal{M}, w) \models \Gamma$ und einem Zustand v mit $(w, v) \in \rho(\underline{while}\ B\ \underline{do}\ \ \alpha\ \underline{end}\ )$. Falls ein solches v nicht existiert ist die gewünschte Aussage $(\mathcal{M}, w) \models [\underline{while}\ B\ \underline{do}\ \ \alpha\ \underline{end}\ ]\,\phi \vee \bigvee \Delta$ trivialerweise wahr.

Aus der Allgemeingültigkeit von $\bigwedge \Gamma \rightarrow Inv$ folgt zunächst $(\mathcal{M}, w) \models Inv$. Die Semantik der while-Schleife garantiert die Existenz von Zuständen $w_0, \ldots, w_k$ mit

$w_0 = w$,
$w_k = v$
$(w_i, w_{i+1}) \in \rho(\alpha)$ für alle $0 \leq i < k$
$(\mathcal{M}, w_i) \models B$ für alle $0 \leq i < k$
$(\mathcal{M}, w_k) \models \neg B$

Daraus und aus Allgemeingültigkeit von $Inv \wedge B \rightarrow [\alpha]\,Inv$ der folgt fr alle $0 \leq i \leq k$ $(\mathcal{M}, w_i) \models Inv$. Die Allgemeingltigkeit der dritten Formel $Inv \wedge \neg B \rightarrow \phi \vee \bigvee \Delta$ liefert jetzt, wie gewnscht, $(\mathcal{M}, v) \models \phi \vee \bigvee \Delta$.

■

3.8.6 Übungsaufgaben

Übungsaufgabe 3.8.1 *In der Definition 87 wird die Semantik einer Formel F durch die Menge $\tau(F)$ aller Welten beschrieben, in denen F wahr ist. Dieselbe Information läßt sich vermitteln durch die Relation $(\mathcal{A}, w) \models F$, wobei*

$$(\mathcal{A}, w) \models F \text{ gdw } w \in \tau(F)$$

Geben Sie eine induktive Definition für $(\mathcal{A}, w) \models F$.

Übungsaufgabe 3.8.2 *Sei eine Struktur $\mathcal{A}$ der dynamischen Aussagenlogik gegeben. Ein Programm α heißt* **deterministisch**, *wenn aus $(s, t_1) \in \rho(\alpha)$ und $(s, t_2) \in \rho(\alpha)$ folgt $t_1 = t_2$. Zeigen Sie, daß für deterministisches α und jede Formel $F \in Fml_{PDL_{reg}}$ gilt*

$$\mathcal{A} \models <\alpha> F \rightarrow [\alpha]F$$

Übungsaufgabe 3.8.3 *Diese Aufgabe soll die Grenzen der Ausdruckstärke der aussagenlogischen Dynamischen Logik PDL_{min} aufzeigen.*

Es gibt keine Formel $F \in Fml_{PDL_{reg}}$, so daß für jede Struktur $\mathcal{A} = (W, \tau, \rho)$ und jeden Zustand $w \in W$ gilt

$(\mathcal{A}, w) \models F$ *gdw* *es gibt keine unendliche Folge $w_0, w_1, \ldots$ von Zusänden $w_i \in W$ mit $(w_i, w_{i+1}) \in \rho^{\mathcal{A}}(a)$ für alle $i \geq 0$.*

wobei a ein atomares Programm ist.

Die genannte Eigenschaft kann z.B. im modalen μ-Kalkül ausgedrückt werden durch $\mu Z.\langle a\rangle Z$.

Übungsaufgabe 3.8.4 *Sie PDL^{-}_{reg} die Erweiterung von PDL_{reg} um einen neuen Operator $^{-}$, der zu jedem Programm α das inverse Programm α^{-} liefert. Genauer heißt das, in jeder Struktur $\mathcal{A} = (W, \tau, \rho)$ für PDL^{-}_{reg} gilt für alle $w_1, w_2 \in W$*

$$(w_1, w_2) \in \rho(\alpha^{-}) \text{ gdw } (w_2, w_1) \in \rho(\alpha)$$

Zeigen Sie, daß auch PDL^{-}_{reg} entscheidbar ist.

Kapitel 4

Temporale Logik

4.1 Lineare Temporale Logik

Faßt man die Welten eines Kripke Rahmens (G, R) als die Zustände eines Weltausschnitts zu verschiedenen Zeitpunkten auf und R als eine zeitliche Vorgängerrelation, so hat man ohne Änderung unserer bisherigen Syntax und Semantik schon eine einfache temporale Logik erhalten. Die Relation $R(s,t)$ wird dabei gelesen als *s liegt zeitlich vor t* und wir schreiben $s < t$ statt $R(s,t)$. Gilt für $\mathcal{K} = (G, <, v)$ und $t \in G$ die Modellbeziehung $(\mathcal{K}, t) \models A$, so sagen wir *die Formel A gilt zum Zeitpunkt t.* Insbesondere wird $(\mathcal{K}, t) \models \Diamond A$ gelesen: *zu einem zukünftigen Zeitpunkt gilt A* und entsprechend $(\mathcal{K}, t) \models \Box A$ als *in jedem zukünftigen Zeitpunkt gilt A*. Wir wollen diesen Bedeutungswechsel auch in der Notation sichtbar machen und schreiben $sif A$ (sometime in the future) anstelle von $\Diamond A$ und $aif A$ (always in the future) für $\Box A$. Was wir bisher einen Rahmen nannten, heißt jetzt eine Zeitstruktur und Kripke Strukturen heißen jetzt temporale Strukturen. Für verschiedene Anwendungsgebiete mögen verschiedene Zeitstrukturen $(G, <)$ interessant sein. Man muß entscheiden, ob die Zeit einen Anfangspunkt haben soll oder sich unendlich in die Vergangenheit erstrecken darf, ob Zeitpunkte diskret aufeinander folgen oder ein dichtes Kontinuum bilden, ob es nur eine Zukunft gibt oder Verzweigungen erlaubt sein sollen. Wir folgen hier der gängigen Klassifizierung in lineare temporale Logiken und verzweigende temporale Logiken (branching time temporal logics).

4.1.1 Einführung

Wir wollen hier zunächst nur die minimale Anforderung stellen, daß $(G, <)$ eine strikte partielle Ordnung ist.
Bei näherer Beschäftigung mit dieser einfachen temporalen Logik stellt man aber entscheidende Mängel fest: man kann in ihr nicht alle zeitlichen Relationen ausdrücken, über die man reden möchte. Es gibt zum Beispiel keine Möglichkeit über die Vergangenheit zu reden. Das läßt sich ändern durch Einführung zusätzlicher temporaler Operatoren, *sip* und *aip* mit der Bedeutung *sometime in*

the past und *always in the past*.

Definition 100 (Vergangenheitsoperatoren)
Für eine temporale Struktur $\mathcal{K} = (G, <, v)$ *und* $t \in G$ *gelte*

$(\mathcal{K}, t) \models sip\ A$	*gdw*	*es gibt ein* $s \in G$ *mit* $s < t$ *und* $(\mathcal{K}, s) \models A$
$(\mathcal{K}, t) \models aip\ A$	*gdw*	*für alle* $s \in G$ *mit* $s < t$ *gilt* $(\mathcal{K}, s) \models A$

Weitere häufig benutzte temporale Operatoren sind die zweistelligen Operatoren *until* und *since*, wobei A *until* B bedeutet, daß irgendwann einmal in der Zukunft B wahr wird und bis dorthin A wahr ist. Dual dazu ist A *since* B wahr, wenn es einen Zeitpunkt in der Vergangenheit gibt, zu dem B wahr war und von da an bis jetzt A gegolten hat.

Definition 101 (until und since)
Sei $\mathcal{K} = (G, <, v)$ *eine temporale Struktur und* $t \in G$.

$(\mathcal{K}, t) \models A\ until\ B$	*gdw*	*es gibt* $s \in G$ *mit* $t < s$ *und* $(\mathcal{K}, s) \models B$ *und*
		für alle r *mit* $t < r < s$ *gilt* $(\mathcal{K}, r) \models A$
$(\mathcal{K}, t) \models A\ since\ B$	*gdw*	*es gibt* $s \in G$ *mit* $s < t$ *und* $(\mathcal{K}, s) \models B$ *und*
		für alle r *mit* $s < r < t$ *gilt* $(\mathcal{K}, r) \models A$

Neben der Infixschreibweise (p *until* q), die wir gewählt haben, findet man auch häufig die Präfixschreibweise $until(q, p)$. Man beachte, daß sich dabei die Reihenfolge der Argumente vertauscht, um im Einklang mit der englischsprachigen Semantik zu bleiben.

Definition 102 (LTL-Formeln)
Die Formeln der linearen Zeitlogik, LTL, (linear temporal logic), werden induktiv wie folgt definiert:

1. *die logischen Konstanten* $true$ *und* $false$ *sind LTL-Formeln,*
2. *jede aussagenlogische Variable ist eine LTL-Formel,*
3. *Sind* F, G *LTL-Formeln, dann auch die aussagenlogischen Kombinationen davon, d.h.* $\neg F$, $F \vee G$, $F \wedge G$, *etc.,*
4. *sind* F, G *LTL-Formeln, dann auch* F *since* G, F *until* G.

Die gerade eingeführten temporalen Operatoren sif, aif, sip und aip kommen in LTL nicht explizit vor, sind aber in LTL definierbar.

Lemma 86 *Die folgenden Äquivalenzen gelten in allen Zeitstrukturen:*

1. $sif\ A \leftrightarrow true\ until\ A$,
2. $aif\ A \leftrightarrow \neg(true\ until\ \neg A)$,
3. $sip\ A \leftrightarrow true\ since\ A$,
4. $aip\ A \leftrightarrow \neg(true\ since\ \neg A)$,

Beweis: Einfach.

∎

Lemma 87 *Die folgenden Äquivalenzen gelten in allen linearen Zeitstrukturen:*

1. $C\ until\ (A \vee B) \leftrightarrow (C\ until\ A) \vee (C\ until\ B)$
2. $C\ since\ (A \vee B) \leftrightarrow (C\ since\ A) \vee (C\ since\ B)$
3. $(B \wedge C)\ until\ A \leftrightarrow (B\ until\ A) \wedge (C\ until\ A)$
4. $(B \wedge C)\ since\ A \leftrightarrow (B\ since\ A) \wedge (C\ since\ A)$

Beweis: Einfach.

∎

Man kann sich eine Vielzahl von Variationen der bisher betrachteten temporalen Operatoren vorstellen und alle wurden wahrscheinlich schon irgendwo einmal definiert. Wir nennen hier nur einige Beispiele. In der Definition von $aif\ A$ wurde der Referenzzeitpunkt ausgeschlossen. Der Operator $fno\ A$ (from now on A) wird durch die Semantikdefinition

$$(\mathcal{K}, t) \models fno\ A \text{ gdw } \text{ für alle } s \in G \text{ mit } t \leq s \text{ gilt } (\mathcal{K}, s) \models A$$

festgelegt. Es hat sich in der Literatur eingebürgert für $fno\ A$ auch $\Box A$ zu schreiben. Entsprechend kann man die Operatoren sif, aip und sip abändern, so daß auch der Referenzzeitpunkt in den Gültigkeitsbereich des All- bzw. Existenzquantors mit ein bezogen wird. Häufig werden auch die Abschwächungen $until_0$ und $since_0$ betrachtet, mit der Semantik:

$(\mathcal{K}, t) \models A\ until_0\ B$	gdw	für alle $s \in G$ mit $t < s$ gilt $(\mathcal{K}, s) \models A$
		oder
		es gibt $s \in G$ mit $t < s$ und $(\mathcal{K}, s) \models B$ und
		für alle r mit $t < r < s$ gilt $(\mathcal{K}, r) \models A$
$(\mathcal{K}, t) \models A\ since_0\ B$	gdw	für alle $s \in G$ mit $s < t$ gilt $(\mathcal{K}, s) \models A$
		oder
		es gibt $s \in G$ mit $s < t$ und $(\mathcal{K}, s) \models B$ und
		für alle r mit $s < r < t$ gilt $(\mathcal{K}, r) \models A$

Fast überflüssig zu sagen, daß alle diese Varianten LTL-definierbar sind.

Eine weitere Variation des *until* Operators besteht in

$(\mathcal{K}, t) \models A\ until^1\ B$	gdw	es gibt $s \in G$ mit $t \leq s$ und $(\mathcal{K}, s) \models B$ und
		für alle r mit $t \leq r < s$ gilt $(\mathcal{K}, r) \models A$

Ein weiterer wichtiger Operator ist der next-Operator, häufig auch mit $\bigcirc$ bezeichnet. Er setzt voraus, daß zu jedem Zeitpunkt t der temporalen Struktur $\mathcal{K}$ ein eindeutig bestimmter nächster Nachfolger existiert. Wir schreiben dann $t+1$ für diesen nächsten Zeitpunkt.

$$(\mathcal{K}, t) \models \bigcirc A \text{ gdw } (\mathcal{K}, t+1) \models A$$

Es gilt die Äquivalenz

$$\bigcirc A \leftrightarrow false\ until\ A.$$

Beachten Sie, daß diese Definition nicht richtig ist für $until^1$ anstelle von *until*.

Definition 103 *Eine Zeitstruktur $\mathcal{K}$, in der zu jedem Zeitpunkt t ein eindutiger nächster Nachfolger existiert heißt eine* diskrete *Zeitstruktur.*

Literaturreferenzen: Wolpers Kapitel [Wolper, 1995] in dem Buch [Bolc & Szałas, 1995]

4.1.2 Ausdrucksstärke

Obwohl man in LTL die meisten vorkommenden temporalen Relationen ausdrücken kann, gibt es auch relativ einfache Aussagen, die in LTL nicht formalisierbar sind. Wir präsentieren hier ein Ergebniss aus [Wolper, 1983].

Lemma 88 *Sei p eine aussagenlogische Variable.*
Für jedes $m \geq 2$ kann die Aussage

Für jeden Zeitpunkt i mit $i = km$, $k \geq 0$ gilt p

durch keine LTL-Formel beschrieben werden.

Beweis Wir benötigen zu diesem Beweis das Resultat aus dem nachfolgenden Lemma 89.

Wir wählen ein festes k, $k \geq 2$. Angenommen es gäbe eine Formel A die in allen temporalen Strukturen $\mathcal{K}$ mit der Zeitstruktur $(\mathbb{N}, <)$ nur dann gilt, wenn für alle Zeitpunkte i mit $i = km$ gilt $(\mathcal{K}, i) \models p$. Sei n die Anzahl der *until*-Operatoren in A. Dann gilt nach Lemma 89 für ein k mit $km - 1 > n$

$$(\mathcal{K}_{km}, 0) \models A \Leftrightarrow (\mathcal{K}_{km-1}, 0) \models A$$

In $\mathcal{K}_{km}$ ist die definierende Forderung für A aber erfüllt und in $\mathcal{K}_{km-1}$ nicht. Widerspruch!

∎

Lemma 89 *Mit $\mathcal{K}_i = (\mathbb{N}, <, v_i)$ bezeichnen wir die temporale Struktur mit $(\mathbb{N}, <)$ als Menge der Zeitpunkte und für $t \in \mathbb{N}$:*

$$v_i(p,t)) = \begin{cases} 1 & \text{falls } t \leq i \\ 0 & \text{falls } t = i+1 \\ 1 & \text{falls } t > i+i \end{cases}$$

Sei A eine beliebige LTL-Formel und n die Anzahl der Vorkommen des until-Operators in A. Dann gilt entweder

$$(\mathcal{K}_i, 0) \models A \text{ für alle } i \geq n+1$$

oder

$$(\mathcal{K}_i, 0) \models \neg A \textit{ für alle } i \geq n+1$$

Mit anderen Worten für $i > n$ *hängt der Wahrheitswert von* $(\mathcal{K}_i, 0) \models A$ *nicht von* i *ab.*

Beweis Der Beweis geschieht durch strukturelle Induktion über die Komplexität der Formel A. Wir unterscheiden folgenden Fälle.

Anfangsfall: $A = p$ Für jedes i gilt $(\mathcal{K}_i, 0) \models p$ genau dann wenn $v_i(p, 0) = 1$ gilt. Das ist aber für alle i der Fall.

1.Induktion Fall: A ist eine aussagenlogische Kombination von A_1, A_2 Einfache Konsequenz aus der Induktionsvoraussetzung.

2.Induktion Fall : $A = B \textit{ until } C$ Wir bemerken zunächst, daß für jede diskreten Zeitstruktur $\mathcal{K}$, und jeden Zeitpunkt z darin gilt:

$$(\mathcal{K}, z) \models B \textit{ until } C \Leftrightarrow (\mathcal{K}, z{+}1) \models C \text{ oder } ((\mathcal{K}, z{+}1) \models B \wedge (B \textit{ until } C)$$

Weiter brauchen wir die Beobachtung, daß für die speziellen Zeitstrukturen $\mathcal{K}_i$, $0 \leq t \leq n$, $i > n$ und vollkommen beliebige Formeln F gilt:

$$(\mathcal{K}_i, t) \models F \text{ gdw } (\mathcal{K}_{i-t}, 0) \models F$$

Jetzt können wir mit dem Beweis des Induktionsschrittes beginnen. Für $i \geq n+1$ erhalten wir durch iterierte Anwendung der ersten Beobachtung

$$\begin{array}{lcl}
(\mathcal{K}_i, 0) \models B \textit{ until } C & \Leftrightarrow & (\mathcal{K}_i, 1) \models C \\
 & & \text{oder} \\
 & & (\mathcal{K}_i, 1) \models B \text{ und } (\mathcal{K}_i, 2) \models C \\
 & & \text{oder} \\
 & & (\mathcal{K}_i, 2) \models B \text{ und } (\mathcal{K}_i, 3) \models C \\
 & & \vdots \text{ oder} \\
 & & (\mathcal{K}_i, i-n) \models B \text{ und } (\mathcal{K}_i, i-n) \models (B \textit{ until } C)
\end{array}$$

Nach der zweiten Beobachtung können wir das äquivalent umformen zu

$$
\begin{array}{lcl}
(\mathcal{K}_i, 0) \models B \textit{ until } C & \Leftrightarrow & (\mathcal{K}_{i-1}, 0) \models C \\
 & & \text{ oder} \\
 & & (\mathcal{K}_{i-1}, 0) \models B \text{ und } (\mathcal{K}_{i-2}, 0) \models C \\
 & & \text{ oder} \\
 & & (\mathcal{K}_{i-2}, 0) \models B \text{ und } (\mathcal{K}_{i-3}, 0) \models C \\
 & & \vdots \text{ oder} \\
 & & (\mathcal{K}_n, 0) \models B \text{ und } (\mathcal{K}_n, 0) \models (B \textit{ until } C)
\end{array}
$$

Nach Induktionsvoraussetzung folgt weiter

$$
\begin{array}{lcl}
(\mathcal{K}_i, 0) \models B \textit{ until } C & \Leftrightarrow & (\mathcal{K}_n, 0) \models C \text{ oder} \\
 & & (\mathcal{K}_n, 0) \models B \text{ und } (\mathcal{K}_n, 0) \models C \\
 & & \text{ oder} \\
 & & (\mathcal{K}_n, 0) \models B \text{ und } (\mathcal{K}_n, 0) \models C \\
 & & \vdots \text{ oder} \\
 & & (\mathcal{K}_n, 0) \models B \text{ und } (\mathcal{K}_n, 0) \models (B \textit{ until } C)
\end{array}
$$

Damit ist gezeigt, daß $(\mathcal{K}_i, 0) \models B \textit{ until } C$ für $i \geq n+1$ nicht von i abhängt.

■

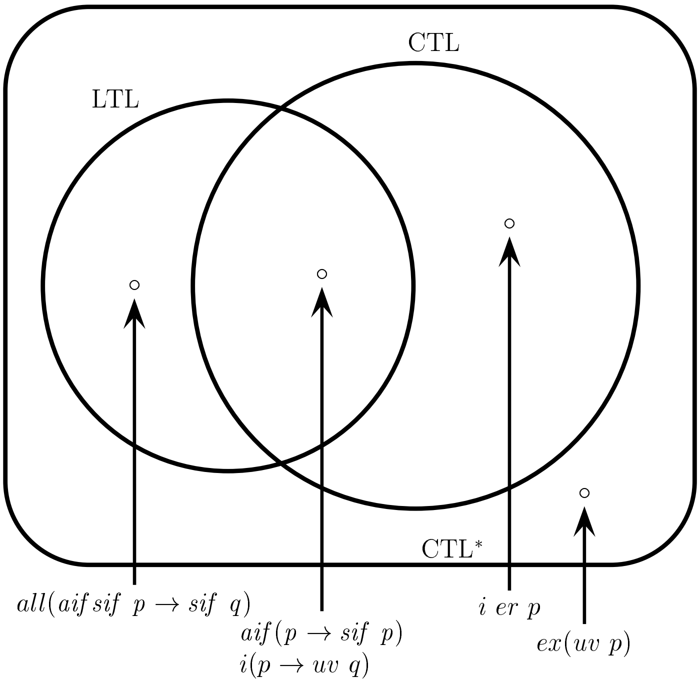


Abbildung 4.1: Vergleich der verschiedenen temporale Logiken

4.1.3 Definierbarkeit

Interessanter ist die Frage, ob es überhaupt einen temporalen Operator gibt, der nicht LTL-definierbar ist. Das ist so noch keine präzise Frage, da wir nicht definiert haben, was ganz allgemein ein temporaler Operator sein soll. Eine Konkretisierung der Frage nach der Definitionsvollständigkeit von LTL erhält man auf dem Umweg über die Reduktion von LTL-Formeln auf prädikatenlogische Formeln. Das Vokabular V_{temp}, in dem die übersetzten Formeln stehen werden, umfaßt dabei für jede Aussagenvariable p ein einstelliges Relationszeichen, das wir wieder mit demselben Buchstaben bezeichen, also $p(x)$ und außerdem ein zweistelliges Relationszeichen $<$ für die Ordnungsrelation.

Definition 104 (Übersetzung in PL1)
Die Übersetzungsfunktion μ von LTL-Formeln in prädikatenlogische Formeln in $Fml(V_{temp})$ wird induktiv definiert durch:

1. $\mu(true) = A(x) \vee \neg A(x)$ *und* $\mu(false) = A(x) \wedge \neg A(x)$ *für ein beliebiges einstelliges Prädikat* $A(x)$*,*
2. $\mu(p) = p(x)$ *für jede Aussagenvariable* p*,*
3. $\mu(\neg F) = \neg\mu(F)$*,* $\mu(F \vee G) = \mu(F) \vee \mu(G)$*,* $\mu(F \wedge G) = \mu(F) \wedge \mu(G)$*, etc.,*
4. $\mu(F \text{ since } G) = \exists y(y < x \wedge \mu(G)(y/x) \wedge \forall z(y < z < x \rightarrow \mu(F)(z/x)))$*,*
5. $\mu(F \text{ until } G) = \exists y(x < y \wedge \mu(G)(y/x) \wedge \forall z(x < z < y \rightarrow \mu(F)(z/x)))$*.*

Man beachte, daß für jede LTL-Formel F die Übersetzung $\mu(F)$ genau x als freie Variable enthält.

Lemma 90 *Für eine temporale Struktur $\mathcal{K} = (G, <, v)$ sei $\mathcal{K}_{temp}$ die prädikatenlogische Struktur, die $(G, <)$ unverändert übernimmt und in der die einstellige Relation $A(s)$ genau dann wahr ist, wenn $v(s, A) = 1$ gilt. Dann gilt für alle LTL-Formeln F und alle $t \in G$*

$$(\mathcal{K}, t) \models F \Leftrightarrow \mathcal{K}_{temp} \models \mu(F)[t]$$

Die Schreibweise $\mu(F)[t]$ soll signalisieren, daß t als Belegung für die (einzige) freie Variable x in $\mu(F)$ benutzt werden soll.

Beweis: Einfaches Nachrechnen.

■

Wir können jetzt den Begriff der Definitionsvollständigkeit von LTL folgendermaßen konkretisieren:

Definition 105 (Definitionsvollständigkeit)
Sei $\mathcal{M}$ eine Klasse von Zeitstrukturen. LTL heißt definitionsvollständig bzgl. $\mathcal{M}$, wenn für jede prädikatenlogisch V_{temp}-Formel $F(x)$ mit genau einer freien Variablen x eine LTL-Formel A existiert, so daß für alle $(G, <) \in \mathcal{M}$ und alle Bewertungsfuntionen v gilt:

$$(G, <, v)_{temp} \models \forall x(\mu(A) \leftrightarrow F(x))$$

Satz 91

1. *LTL ist definitionsvollständig für die Zeitstruktur $(Z, <)$ der ganzen Zahlen.*
2. *LTL ist definitionsvollständig für die Zeitstruktur $(\mathbb{R}, <)$ der reellen Zahlen.*
3. *LTL ist nicht definitionsvollständig für die Zeitstruktur $(Q, <)$ der rationalen Zahlen.*
4. *LTL ist nicht definitionsvollständig für die Klasse $\mathcal{M}_{lin}$ aller linearer Zeitstrukturen.*

Beweis: Die Beweise sind zum Teil sehr kompliziert. Die beste Referenz dafür ist [Gabbay *et al.*, 1994]. Die Pionierarbeit auf dem Gebiet der Definierbarkeit temporaler Operatoren ist [Kamp, 1968]. Um einen Eindruck zu vermitteln von der Art der Beweisführung und den dabei zu lösenden Teilproblemen wollen wir wenigstens den einfachsten positiven Fall der Definitionsvollständigkeit über den ganzen Zahlen genauer betrachten. Wir folgen dabei der Darstellung in [Gabbay *et al.*, 1994]. Wir sammeln zunächst noch einige Äquivalenzen, die später nützlich sein werden.

Lemma 92 *Die folgenden Äquivalenzen gelten in der Zeitstruktur der ganzen Zahlen:*

1. $\neg(B\ until\ A) \leftrightarrow (aif\ \neg A) \vee (\neg A\ until\ (\neg A \wedge \neg B))$
2. $\neg(B\ since\ A) \leftrightarrow (aip\ \neg A) \vee (\neg A\ since\ (\neg A \wedge \neg B))$
3. $\neg(B\ until\ A) \leftrightarrow (aif\ \neg A) \vee ((\neg A \wedge B)\ until\ (\neg A \wedge \neg B))$
4. $\neg(B\ since\ A) \leftrightarrow (aip\ \neg A) \vee ((\neg A \wedge B)\ since\ (\neg A \wedge \neg B))$

Beweis: Es genügt 3. und 4. zu beweisen, da 1. und 2 Spezialfälle davon sind. Wir beweisen 3. Nach Definition ist $t \models \neg(B\ until\ A)$ äquivalent zu

$$\text{Für alle } s > t \text{ mit } s \models A \text{ gibt es ein } s', s > s' > t \text{ mit } s' \models \neg B$$

Wir wollen uns überlegen, daß daraus die rechte Seite von 3. folgt. Falls $aif\ \neg A$ gilt, sind wir schon fertig. Anderenfalls gibt es ein $s > t$ mit $s \models A$. Wir wählen das kleinste solche s, d.h. für alle s' mit $s > s' > t$ gilt $s' \models \neg A$. Nach Voraussetzung gibt es ein s' mit $s > s' > t$ mit $s' \models \neg B$. Wir wählen wieder das kleinste solche s'. Somit gilt $s' \models \neg A \wedge \neg B$ und für alle t' mit $s' > t' > t$ gilt $t' \models B \wedge \neg A$. Das heißt aber es gilt $t \models aif\ \neg A \vee ((\neg A \wedge B)\ until\ (\neg A \wedge \neg B))$.
Gelte nun umgekehrt die rechte Seite von 3. zum Zeitpunkt t. Falls $t \models aif\ \neg A$ gilt,dann gilt offensichtlich auch $t \models \neg(B\ until\ A)$. Anderenfalls gibt es ein $s > t$ mit $s \models (\neg A \wedge \neg B)$ und für alle s' mit $s > s' > t$ gilt $s' \models (\neg A \wedge B)$. Wir zeigen, daß daraus die linke Seite von 3. folgt. Sei dazu ein $u > t$ gegeben mit $u \models A$, dann muss $u > s$ gelten und deswegen gibt es $u > s' > t$ mit $s' \models \neg B$.
Teil 4. des Lemmas wird analog bewiesen.

∎

Die wesentliche Idee für die Beweise der Definitionsvollständigkeit, die wir im folgenden betrachten werden, ist die Einführung des Begriffes einer trennbaren Logik. Dieser Begriff liegt nicht an der Oberfläche, er bietet sich nicht offensichtlich an, sondern durch eine gründliche Analyse des Problems, wurde zu Tage gefördert, daß eine temporale Logik genau dann definitionsvollständig ist, wenn sie die Trennbarkeitseigenschaft besitzt. Der Beweis dieses Zusammenhangs bildet den ersten Schritt in unserem Beweis. Der zweite Schritt besteht nun darin für eine gegebene temporale Aussagenlogik TL und eine Klasse von Zeitstrukturen $\mathcal{M}$ die Trennbarkeitseigenschaft nachzuweisen. Jetzt, zu den Details!

Definition 106 (Zukunfts- und Vergangenheitsformeln) *Sei TL eine beliebige aussagenlogische Temporallogik, $\mathcal{M}$ eine Klasse von Zeitstrukturen.*
Eine Formel A aus TL heißt eine Zukunftsformel *bzgl. $\mathcal{M}$, wenn für jede Zeitstruktur $(G, <)$ in $\mathcal{M}$, jeden Zeitpunkt $t \in G$ und je zwei Interpretationsfunktionen v_1, v_2 mit $v_1(s) = v_2(s)$ für alle $s > t$ gilt*

$$(G, <, v_1, t) \models A \text{ gdw } (G, <, v_2, t) \models A$$

Eine Formel A aus TL heißt eine Vergangenheitsformel *bzgl. $\mathcal{M}$, wenn für jede Zeitstruktur $(G, <)$ in $\mathcal{M}$, jeden Zeitpunkt $t \in G$ und je zwei Interpretationsfunktionen v_1, v_2 mit $v_1(s) = v_2(s)$ für alle $s < t$ gilt*

$$(G, <, v_1, t) \models A \text{ gdw } (G, <, v_2, t) \models A$$

Knapp gesagt ist A eine Zukunftsformel, wenn ihre Gültigkeit zum Zeitpunkt t nur von der Zukunft, d.h. nur von den $s > t$ abhängt.

Beispiel 23 *Die Formeln*

$$sif\ p, aif\ p, (p \text{ until } q), (p \text{ until}(q \text{ until } r))$$

sind Zukunftsformeln,

$$sip\ p, aip\ p, (p \text{ since } q), (p \text{ since}(q \text{ since } r))$$

sind Vergangenheitsformeln und

$$(p \text{ until}(q \text{ since } r)), (p \text{ since}(q \text{ until } r))$$

sind weder Zukunfts- noch Vergangenheitsformeln.

Definition 107 (Trennbarkeit)
Eine aussagenlogische Temporallogik TL ist trennbar *bzgl.* $\mathcal{M}$*, wenn jede Formel A über* $\mathcal{M}$ *äquivalent ist zu einer Booleschen Kombination von Zukunftsformeln, Vergangenheitsformeln und atomaren Formeln.*

Satz 93 *Sei TL eine temporale Aussagenlogik,* $\mathcal{M}$ *eine Klasse linearer Zeitstrukturen, so daß mindestens die Operatoren* sif *und* sip *in TL über* $\mathcal{M}$ *definierbar sind und TL trennbar ist bzgl.* $\mathcal{M}$*. Dann ist TL definitionsvollständig für* $\mathcal{M}$*.*

Beweis: Machen wir uns noch einmal klar, was bewiesen werden muß. Zu jeder Formel $F(x)$ der Prädikatenlogik erster Stufe über dem Vokabular V_{temp} mit höchstens x als freier Variablen gibt es eine Formel A der temporalen Logik TL so daß für jede temporale Struktur $\mathcal{K} = (G, <, v)$ mit $(G, <) \in \mathcal{M}$ und jeden Zeitpunkt $t \in G$ gilt

$$(\mathcal{K}, t) \models A \text{ gdw } \mathcal{K}_{temp} \models F(t)$$

Diese Korrespondenz wird durch Induktion über die Quantorentiefe von $F(x)$ bewiesen. Ist $F(x)$ quantorenfrei, dann ist x die einzige Variable, die in F vorkommt. $F(x)$ ist also eine Boolesche Kombination von atomaren einstelligen Formeln $Q(x)$ und Formeln der Form $x = x$, $x < x$. Die letzten beiden kann man eliminieren, indem man sie durch die Konstanten t bzw. f ersetzt. Die temporale Formel A erhält man jetzt durch Ersetzung der einstelligen Prädikate $Q(x)$ durch aussagenlogische Variablen Q. Soweit, so einfach. Wir kommen jetzt zum **Induktionsschritt**: $F(x) = \exists y H(x, y)$ Wir nehmen an, daß die Variable x in $F(x)$ nicht quantifiziert vorkommt und Teilformeln $x = x$ und $x < x$,wie oben, eliminiert wurden. Wir können die gewünschte Äquivalenz nicht in einem Schritt erreichen und werden zunächst ein etwas schwächeres Resultat beweisen. Als Vorbereitung dazu erweitern wir das Vokabular V_{temp} vorübergehend zu V^0_{temp} durch Hinzunahme der einstelligen Relationszeichen $R_=, R_<$ und $R_>$. Sei $H^0(x, y)$ die Formel, die aus $H(x, y)$ hervorgeht, indem für alle Variablen z alle Teilformeln $x = z$ durch $R_=(z)$, alle Teilformeln $x < z$ durch $R_<(z)$ und alle $x > z$ durch $R_>(z)$ ersetzt werden.

Für eine V_{temp}-Struktur $\mathcal{K}_{temp}$ mit Universum G und $t \in G$ definiert man eine V^0_{temp}-Struktur $\mathcal{K}^t_{temp}$ die V^0_{temp}-Struktur durch die Festlegung:

$$
\begin{array}{lll}
\mathcal{K}^t_{temp} \models R_=(s) & \text{gdw} & s = t \\
\mathcal{K}^t_{temp} \models R_<(s) & \text{gdw} & s < t \\
\mathcal{K}^t_{temp} \models R_>(s) & \text{gdw} & s > t
\end{array}
$$

Dann gilt für alle $s \in G$

$$\mathcal{K}_{temp} \models H(t,s) \text{ gdw } \mathcal{K}^t_{temp} \models H^0(t,s),$$

Die Variable x kommt in H^0 nur noch als Argument einstelliger Prädikate aus V_{temp} vor. Somit läßt sich H^0 durch elementare Umformungen in eine logisch äquivalente Formel der Form

$$\bigvee_j [H_j(x) \wedge G^0_j(y)]$$

transformieren, wobei

- jedes $H_j(x)$ eine Boolesche Kombination einstelliger Prädikatszeichen aus dem nicht erweiterten Vokabular V_{temp} ist,
- jedes $G^0_j(y)$ eine Formel von geringerer Quantorentiefe als $F(x)$ ist und höchstens y als freie Variable enthält.

Nach Induktionsvoraussetzung gibt es TL-Formeln C_j so daß für jede temporale Struktur $\mathcal{H} = (G, <, v)$ mit $(G, <) \in \mathcal{M}$ und jeden Zeitpunkt $s \in G$ gilt

$$(\mathcal{H}, s) \models C_j \text{ gdw } \mathcal{H}_{temp} \models G^0_j(s)$$

Offensichtlich gilt dann für jedes $t \in G$

$$(\mathcal{H}, t) \models sif\ C_j \vee C_j \vee sip\ C_j \text{ gdw } \mathcal{H}_{temp} \models \exists y G^0_j(y)$$

Da auch die atomaren Formeln $H_j(x)$ durch TL-Formeln ausgedrückt werden können und der Existenzquantor sich in dem vorliegenden Fall über Disjunktionen und Konjunktionen nach innen ziehen läßt erhalten wir insgesamt eine TL-Formel C, so daß für jede temporale Struktur $\mathcal{H} = (G, <, v)$ mit $(G, <) \in \mathcal{M}$ und jeden Zeitpunkt $t \in G$ gilt

$$(\mathcal{H}, t) \models C \text{ gdw } \mathcal{H}_{temp} \models \exists y H^0(t, y)$$

Das ist noch nicht das gewünschte Resultat, da im Vokabular V^0_{temp} von $H^0(x, y)$ und $\mathcal{H}_{temp}$ die zusätzlichen einstelligen Prädiakte $R_=, R_>$ und $R_<$ vorkommen und entsprechend aussagenlogische Variable $r_=, r_>$ und $r_<$ im Vokabular V^0 von C und $\mathcal{H}$. Im zweiten Teil des Beweises werden wir zeigen, wie wir diese zusätzlichen Symbole wieder los werden können und dazu brauchen wir die Trennbarkeitseigenschaft von TL bzgl. $\mathcal{M}$. Danach gibt es eine zu C bzgl $\mathcal{M}$ äquivalente TL-Formel der Form

$$\bigvee_j [C^v_j \wedge C^z_j \wedge C^=_j]$$

wobei die C^v_j alle reine Vergangenheitsformeln, die C^z_j reine Zukunftsformeln und die $C^=_j$ negierte oder unnegierte Aussagenvariablen sind. Die TL-Formeln B^v_j entstehen, indem in C^v_j die temporale Aussagenvariable $r_>$ durch die logische Konstante $true$ und die Variablen $r_<$ und $r_=$ durch $false$ ersetzt werden. Analog entstehe B^z_j aus C^z_j, indem $r_<$ durch $true$ und $r_>$ und $r_=$ durch $false$ ersetzt werden. Schließlich geht $B^=_j$ aus $C^=_j$ hervor durch die Ersetzung von $r_=$ durch $true$ und $r_>$ und $r_<$ durch $false$. Die Formel

$$B = \bigvee_j [B^v_j \wedge B^z_j \wedge B^=_j]$$

ist dann eine TL-Formel im ursprünglichen Vokabular V. Wir zeigen, daß für unsere Zwecke C durch B ersetzt werden kann. Sei $v : V \times G \to \{0, 1\}$ eine Interpretation auf dem nicht erweiterten Vokabular und die Erweiterung von v zu $v^t : V^0 \times G \to \{0, 1\}$ sei durch

$$\begin{array}{lcl} v^t(r_>, s) = 1 & \leftrightarrow & t > s \\ v^t(r_<, s) = 1 & \leftrightarrow & t < s \\ v^t(r_=, s) = 1 & \leftrightarrow & t = s \end{array}$$

festgelegt.
Wir betrachten eine zweite Interpretationsfunktion $v^v : V^0 \times G \rightarrow \{0,1\}$, die durch

$$\begin{array}{ll} v^v(r_>, s) = 1 & \text{für alle } s \\ v^v(r_<, s) = 0 & \text{für alle } s \\ v^v(r_=, s) = 0 & \text{für alle } s \end{array}$$

Offensichtlich stimmen v^v und v^t auf der Vergangenheit von t überein, weswegen

$$(G, v^t, t) \models C_j^v \text{ gdw } (G, v^v, t) \models C_j^v$$

gilt. Nach Definition von v^v gilt aber auch

$$(G, v^v, t) \models C_j^v \text{ gdw } (G, v^v, t) \models B_j^v$$

Da in B_j^v aber nur noch temporale Variable aus V vorkommen haben wir insgesamt die Äquivalenz

$$(G, v^t, t) \models C_j^v \text{ gdw } (G, v, t) \models B_j^v$$

Analog zeigt man

$$(G, v^t, t) \models C_j^z \text{ gdw } (G, v, t) \models B_j^z$$

und

$$(G, v^t, t) \models C_j^= \text{ gdw } (G, v, t) \models B_j^=$$

also ingesamt

$$(G, v^t, t) \models C \text{ gdw } (G, v, t) \models B$$

Ziehen wir auch den ersten Teil des Beweises in diese Zusammenfassung mit ein, so erhalten wir:

$$\begin{array}{lll} \mathcal{K}_{temp} \models F(t) & \text{gdw} & \mathcal{K}_{temp} \models \exists y H(t,y) \\ & \text{gdw} & \mathcal{K}_{temp}^t \models \exists y H^0(t,y) \\ & \text{gdw} & (\mathcal{K}^t, t) \models C \\ & \text{gdw} & (\mathcal{K}, t) \models B \end{array}$$

■

Satz 93 besagt, daß unter den genannten Voraussetzungen die Trennbarkeit einer temporalen Logik eine hinreichende Bedingung für ihre Definitionsvollständigkeit ist. Die Trennbarkeit ist aber auch eine notwendige Bedingung.

Satz 94 *Sei TL eine temporale Aussagenlogik, die definitionsvollständig ist für die Klasse $\mathcal{M}$ von Zeitstrukturen*
Dann besitzt TL die Trennbarkeitseigenschaft bzgl. $\mathcal{M}$.

Beweis: siehe [Gabbay *et al.*, 1994], Theorem 9.3.4, p. 36.1

■

In dem folgenden Beweis der Trennbarkeitseigenschaft von LTL über den ganzen Zahlen benutzen wir eine syntaktische Definition für die Trennbarkeit.

Definition 108 (Syntaktisch separierte Formeln)

1. *Eine Formel der Form $(A \text{ until } B)$, so daß weder in A noch in B der Operator since auftritt heißt eine* **syntaktische Zukunftsformel**.
2. *Eine Formel der Form $(A \text{ since } B)$, so daß weder in A noch in B der Operator until auftritt heißt eine* **syntaktische Vergangenheitsformel**.
3. *Eine Formel heißt* **syntaktisch separiert**, *wenn sie eine Boolesche Kombination von atomaren Formeln, syntaktischen Zukunfts- und syntaktischen Vergangenheitsformeln ist.*

Lemma 95

1. *Eine syntaktische Zukunftsformel ist eine Zukunftsformel.*
2. *Eine syntaktische Vergangenheitsformel ist eine Vergangenheitsformel.*

3. *Falls jede Formel einer aussagenlogischen Temporallogik TL bezüglich einer Klasse $\mathcal{M}$ von Zeitstrukturen äquivalent zu einer syntaktisch separierten Formel ist, dann besitzt TL die Trennbarkeitseigenschaft.*

Beweis: Die Behauptungen 1. und 2. beweist man durch Induktion über die Komplexität der Formeln A und B. Teil 3. ist eine unmittelbare Konsequenz aus 1. und 2.

∎

Der Beweis der Trennbarkeit von LTL wird schließlich durch ein induktives Argument geführt werden. Die kritischen Induktionsschritte werden durch die folgenden drei Lemmas gegeben.

Lemma 96 *Für beliebige Formeln A, B, C, D gelten für die Zeitstruktur der ganzen Zahlen die folgenden Äquivalenzen:*

1. $D\ \mathit{since}\ (C \wedge (B\ \mathit{until}\ A)) \quad \Leftrightarrow$
   $(D\ \mathit{since}\ C) \wedge (B\ \mathit{since}\ C) \wedge B \wedge (B\ \mathit{until}\ A)$
   $\vee$
   $(D\ \mathit{since}\ C) \wedge (B\ \mathit{since}\ C) \wedge A$
   $\vee$
   $D\ \mathit{since}\ (A \wedge D \wedge (D\ \mathit{since}\ C) \wedge (B\ \mathit{since}\ C))$
2. $(C \vee (B\ \mathit{until}\ A))\ \mathit{since}\ D \quad \Leftrightarrow$
   $C\ \mathit{since}\ D$
   $\vee$
   $(\Phi\ \mathit{since}\ D) \wedge B \wedge (B\ \mathit{until}\ A)$
   $\vee$
   $(\Phi\ \mathit{since}\ D) \wedge A$
   $\vee$
   $C\ \mathit{since}(A \wedge (\Phi\ \mathit{since}\ D))$
   wobei
   $\Phi = B \vee (C\ \mathit{since}\ D) \vee (C\ \mathit{since}\ (A \wedge C)) \vee A$
3. $D\ \mathit{until}\ (C \wedge (B\ \mathit{since}\ A)) \quad \Leftrightarrow$
   $(D\ \mathit{until}\ C) \wedge (B\ \mathit{until}\ C) \wedge B \wedge (B\ \mathit{since}\ A)$
   $\vee$
   $(D\ \mathit{until}\ C) \wedge (B\ \mathit{until}\ C) \wedge A$
   $\vee$
   $D\ \mathit{until}\ (A \wedge D \wedge (D\ \mathit{until}\ C) \wedge (B\ \mathit{until}\ C))$

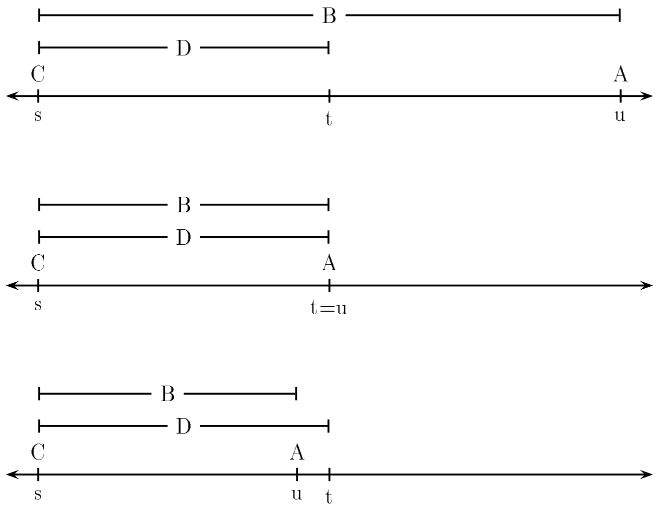


Abbildung 4.2: Drei Möglichkeiten für $t \models D \; since \; (C \wedge (B \; until \; A))$

4. $(C \vee (B \; since \; A)) \; until \; D \quad \Leftrightarrow$
$C \; until \; D$
$\vee$
$(\Phi \; until \; D) \wedge B \wedge (B \; since \; A)$
$\vee$
$(\Phi \; until \; D) \wedge A$
$\vee$
$C \; until(A \wedge (\Phi \; until \; D))$
wobei
$\Phi = B \vee (C \; until \; D) \vee (C \; until \; (A \wedge C)) \vee A$

Beweis:
zu 1:

Gilt zum Zeitpunkt t die Formel $D \; since \; (C \wedge (B \; until \; A))$, dann gibt es einen Zeitpunkt $s < t$ und einen Zeitpunkt $u > s$ mit $s \models C$, $u \models A$, für alle $s < s' < t$ gilt $s' \models D$ und für alle $s < u' < u$ gilt $u' \models B$. Wir können drei Fälle unterscheiden, je nach der Lage des Punktes u: $u > t$, $u = t$ und $u < t$, siehe Abbildung 4.2. Man sieht sofort, daß im ersten Fall die erste Disjunktion der Formel auf der rechten Seite von $\Leftrightarrow$ gilt, im zweiten Fall die

zweite Disjunktion und im dritten die dritte.
Gelte jetzt die rechte Seite der Äquivalenz. Wir betrachten den Fall, daß die erste Disjunktion gilt:

$$t \models (D \; since \; C) \wedge (B \; since \; C) \wedge B \wedge (B \; until \; A)$$

Dann gibt es $s_1 < t$ und $s_2 < t$ mit $s_1 \models C$, $s_2 \models C$, für alle $s_1 < s' < t$ gilt $s' \models D$ und für alle $s_2 < s' < t$ gilt $s' \models B$. Wegen der Linearität der betrachteten Zeitstruktur gibt es auch ein $s < t$ (z.B. $s = \max\ (s_1, s_2)$) mit $s \models C$ und für alle $s < s' < t$ gilt $s' \models (D \wedge B)$. Da auch $t \models B \wedge (B \; until \; A)$ gilt gibt es ein $t < u$ mit $u \models A$ und für alle $t \leq u' < u$ gilt $u' \models B$. Daraus folgt $s \models B \; until A$ ingesamt also $t \models D \; since \; (C \wedge (B \; until \; A))$. Die Beweise im Falle, daß die zweite oder dritte Disjunktion gilt, verlaufen analog.

zu 2:

Gelte $t \models (C \vee (B \; until \; A)) \; since \; D$. Dann gibt es $s < t$ mit $s \models D$ und für alle $s < s' < t$ gilt $s' \models C \vee (B \; until \; A)$. Gilt für alle $s < s' < t$ $s' \models C$, dann gilt $t \models (C \; since \; D)$ und wir sind fertig. Anderenfalls gibt es einen Zeitpunkt $s < s_1$, zu dem zum ersten Mal $\neg C$ wahr wird. Jetzt muß $s_1 \models (B \; until \; A)$ gelten. Es gibt also einen Zeitpunkt $s_1 < u_1$ mit $u_1 \models A$ und für alle $s_1 < s' < u_1$ gilt $s' \models B \wedge \neg A$. u_1 ist hier als der erste Zeitpunkt oberhalb s_1 gewählt, an dem A gilt. Falls $u_1 < t$ ist beginnt das ganze von vorne: entweder gilt $s' \models C$ für alle $u_1 < s' < t$ oder es gibt ein s_2 nach u_1 an dem zum ersten Mal wieder $\neg C$ gilt, usw. Es kann hier auch $s_2 = u_1$ sein. Zusammenfassend, gibt es also Zeitpunkte $s_1, \ldots, s_k$ und $u_1, \ldots, u_k$ mit

$$s < s_1 < u_1 \leq s_2 < u_2 \leq \ldots \leq s_k < u_k$$

und

$$\begin{array}{rlll}
\text{für alle } i & s_i & \models & \neg C \\
\text{für alle } s < s' < s_1 & s' & \models & C \\
\text{für alle } s_i < s' < u_i & s' & \models & B \wedge \neg A \\
\text{für alle } i & u_i & \models & A \\
\text{für alle } u_i \leq s' < s_{i+1} & s' & \models & C
\end{array}$$

Wir unterscheiden wieder drei Fälle, je nachdem ob $t < u_k$, $t = u_k$ oder $u_k < t$ gilt. Diesen Fällen entsprechen auch in derselben Reihenfolge die Disjunktionsglieder 2 bis 4 der Formel auf der rechten Seite der Äquivalenz. Wir betrachten ausführlich den dritten Fall und überlassen die einfachen Anpassungen für die Beweise der restlichen Fälle dem Leser.
Im dritten Fall gilt $s_k < t < u_k$ und daraus folgt nach der Wahl der Zeitpunkte s_i und u_i sofort $t \models B$ und $t \models B\ until\ A$. Es bleibt zu zeigen, daß für jeden Zeitpunkt $s < t' < t$ die Aussage $t' \models \Phi$ gilt, deren Definition wir zur Bequemlichkeit des Lesers noch einmal wiederholen:

$$\Phi = B \vee (C\ since\ D) \vee (C\ since\ (A \wedge C)) \vee A$$

Für $s < t' \leq s_1$ gilt $t' \models C\ since\ D$ und damit auch $t' \models \Phi$. Für $s_i < t' < u_i$ gilt die erste Disjunktion von Φ, also $t' \models B$. Für alle $i \geq 1$ gilt $u_i \models A$, ebenfalls ein Disjunkt von Φ. Falls $u_i < s_{i+1}$ ist gilt für alle $u_i < t' < s_{i+1}$ $t' \models C\ since\ (A \wedge C)$. Hierzu beachte man, daß im Fall $u_i \models \neg C$ nach der Wahl der Zeitpunkte $u_i = s_{i+1}$ sein muss.

Gelte jetzt umgekehrt $t \models (\Phi\ since\ D) \wedge B \wedge (B\ until\ A)$. Wir wollen zeigen, daß daraus $t \models (C \vee (B\ until\ A))\ since\ D$ folgt. Es gibt also Zeitpunkte $s < t$ und $t < u$ mit $s \models D$, $u \models A$, für alle $t \leq u' < u$ gilt $u' \models B$ und für alle $s < s' < t$ gilt $s' \models \Phi$. Wir können außerdem s so wählen, daß für alle $s < s' < t$ gilt $s' \models \neg D$. Wir wollen für alle $s < s' < t$ zeigen, daß $s' \models C \vee (B\ until\ A)$ gilt. Angenommen das trifft nicht zu und es gibt einen Zeitpunkt s' in dem angegebenen Intervall mit $s' \models \neg C \wedge \neg(B\ until\ A)$. Wir wissen schon, daß es einen Zeitpunkt $u > t > s'$ mit $u \models A$ gibt, daran kann also $s' \models B\ until\ A$ nicht scheitern. Aus $s' \models \neg(B\ until\ A)$ folgt die Existenz eines Zeitpunktes $t' < t$, so daß $t' \models \neg B \wedge \neg A$ und für alle $s' < r < t'$ gilt $r \models B \wedge \neg A$. Wir überlegen uns, daß $t' \models \Phi$ nicht gelten kann, was dann den gewünschten Widerspruch ergibt. Ganz offensichtlich sind der erste und der letzte Disjunkt von Φ zum Zeitpunkt t' falsch. Aber auch $t' \models C\ since\ D$ kann nicht zutreffen, da nach Wahl von s jeder Zeitpunkt $s_0 < t'$, der die Formel D erfüllt auch $s_0 \leq s$ erfüllen muß. Wegen $s_0 < s' < t'$ mit $s' \models \neg C$

folgt $t' \models \neg(C\ since\ D)$. Es bleibt die letzte Möglichkeit $t' \models C\ since\ (A \wedge C)$ zu untersuchen. Aber der früheste Zeitpunkt vor t', der $(A \wedge C)$ erfüllen kann, liegt auf jeden Fall strikt vor s', was auch diese Möglichkeit ausschließt. Die Annahme $s' \models \neg C \wedge \neg(B\ until\ A)$ ist somit zum Widerspruch geführt worden.

zu 3. und 4:

analog zu 1. und 2.

∎

Lemma 97 *Für beliebige Formeln A, B, C, D gelten für die Zeitstruktur der ganzen Zahlen die folgenden Äquivalenzen:*

1. $D\ since\ (C \wedge \neg(B\ until\ A))$ $\Leftrightarrow$
   ?
2. $(C \vee \neg(B\ until\ A))\ since\ D$ $\Leftrightarrow$
   ?
3. $D\ until\ (C \wedge \neg(B\ since\ A))$ $\Leftrightarrow$
   ?
4. $(C \vee \neg(B\ since\ A))\ until\ D$ $\Leftrightarrow$
   ?

Lemma 98 *Für beliebige Formeln A, B, C, D gelten für die Zeitstruktur der ganzen Zahlen die folgenden Äquivalenzen:*

1. $(D \vee (B\ until\ A))\ since\ (C \wedge \neg(B\ until\ A))$ $\Leftrightarrow$
   ?
2. $(C \vee \neg(B\ until\ A))\ since\ (D \wedge (B\ until\ A))$ $\Leftrightarrow$
   ?
3. $(D \vee (B\ since\ A))\ until\ (C \wedge \neg(B\ since\ A))$ $\Leftrightarrow$
   ?
4. $(C \vee \neg(B\ since\ A))\ until\ (D \wedge (B\ since\ A))$ $\Leftrightarrow$
   ?

Um zu zeigen, daß durch wiederholte Anwendung von Lemma 96, 97 und 98 eine Formel in eine äquivalente syntaktisch separierte Formel transformierte werden kann, brauchen wir ein Komplexitätsmaß für LTL-Formeln.

Definition 109 *Für eine beliebige LTL-Formel A definieren wir die* **until-Komplexität** $uk(A)$, *bzw. die* **since-Komplexität** $sk(A)$ *wie folgt:*

1. $uk(A) = \#(um(A))$,
   wobei $um(A) = \{(B \textit{ until } C) \mid (B \textit{ until } C) \textit{ ist Teilformel in } A\}$
2. $sk(A) = \#(sm(A)$,
   wobei $sm(A) = \{(B \textit{ since } C) \mid (B \textit{ since } C) \textit{ ist Teilformel in } A\}$

Es ist wichtig zu bemerken, daß eine Formel $(B \textit{ until } C)$, die in A mehrfach vorkommt, für $uk(A)$ nur einmal gezählt wird.

Lemma 99 *Sind A,B syntaktisch separierte Formeln, dann gibt es auch äquivalente syntaktisch separierte Formeln R_1, bzw R_2 zu*

1. $A \textit{ since } B$
2. $A \textit{ until } B$

Außerdem können R_1 und R_2 so gewählt werden, daß

1. $uk(R_1) = uk(A \textit{ since } B)$
2. $sk(R_2) = sk(A \textit{ until } B)$

gilt.

Beweis:
Der Beweis von 1. wird durch Induktion über $uk = uk(A)+uk(B)$ geführt. Falls $uk = 0$ ist, dann ist $(A \textit{ since } B)$ schon syntaktisch separiert. Sei jetzt $uk > 0$. Sei $B = \bigvee_j B_j$ eine disjunktive Normalform von B und $A = \bigwedge A_k$ eine konjunktive Normalform von A. Nach Lemma 87 genügt es für jedes j und k zu zeigen, daß

(A_k *since* B_j) in eine äquivalente syntaktisch separierte Formel transformiert werden kann. Jedes A_k und jedes B_j ist von der Form

$$\begin{array}{rl} C_0 \wedge & (D_1 \ until\ E_1) \wedge \ldots \wedge (D_n \ until`E_n) \\ \wedge & \neg(D_{n+1} \ until\ E_{n+1}) \wedge \ldots \wedge \neg(D_m \ until\ E_m) \end{array}$$

Hierbei ist C_0 eine Konjunktion aussagenlogischer Literale. Wir wählen ein (D_i *until* E_i)), das nicht als echte Teilformel in einem (D_r *until* E_r) vorkommt. Da die Teilformelrelation keine Zyklen haben kann, muß es ein solches (D_i *until* E_i)) geben. Wir können außerdem annehmen, daß innerhalb A_k bzw. innerhalb B_j die Teilformel (D_i *until* E_i)) nur einmal vorkommt, denn doppelte Vorkommen mit dem gleichen Vorzeichen können äquivalent zu einem Vorkommen zusammengefaßt werden. Kommt dagegen (D_i *until* E_i)) einmal negiert und einmal unnegiert in A_k vor, dann ersetzen wir ganz A_k durch $true$, tritt diese Situation in B_j auf, so ersetzen wir B_j durch $false$. Es bleiben also insgesamt 6 Fälle zu untersuchen, die den jeweils ersten beiden Teilen der Lemmas 96, 97 und 98 entsprechen.
Wir betrachten ausführlich den Fall, daß (D_i *until* E_i)) unnegiert in einem B_j vorkommt. Losgelöst von allen Indizes besteht die Aufgabe darin, zu $F = (A\ since(C \wedge (D\ until\ E))$ eine äquivalente syntaktisch separierte Formel zu finden, wobei D *until* E) in A und C nicht vorkommt und D und E schon syntaktisch separiert sind. Wir können außerdem Gebrauch machen von der Induktionsvoraussetzung, daß alle Formeln mit until-Komplexität $< uk(F)$ äquivalent in syntaktisch separierte Formeln transformiert werden können. Nach Lemma 96 Teil 1 können wir F umschreiben in eine Boolesche Kombination von folgenden Formeln:

1. $A\ since\ C$
2. $D\ since\ C$
3. D
4. $D\ until\ E$
5. E
6. $A\ since\ (E \wedge (A\ since\ C) \wedge (D\ since\ C))$

Nach Wahl von $(D\ until E)$ ist die until-Komplexität all dieser Formeln echt kleiner als $uk(F)$. Eine Ausnahme davon macht möglicherweise Formel 4, aber die ist sowieso schon syntaktisch separiert. Die Formeln 1 - 5 können somit in äquivalente syntaktisch separierte Formeln transformiert werden. Formel 6 hat zwar kleinere until-Komplexität als $uk(F)$, aber die rechte Seite des übergeordneten *since*-Operators ist nicht mehr notwendigerweise syntaktisch separiert - in E und D können noch weitere *until*-Operatoren vrokommen. Die Induktionsvoraussetzung erlaubt es uns jedoch die rechte Seite

$$(E \wedge (A\ since\ C) \wedge (D\ since\ C))$$

in eine äquivalente syntaktisch transformierte Formel R umzuwandeln. In einer weiteren Anwendung der Induktionsvoraussetzung können wir jetzt $A\ since\ R$ syntaktisch separieren. An der Stelle müssen wir aber von der zusätzlichen Information gebrauch machen, daß

$$mk(R) = mk(E \wedge (A\ since\ C) \wedge (D\ since\ C))$$

gilt. Da die Formeln 1 - 6 nur aus Unterformeln von F aufgebaut sind und keine neuen *until*-Formeln einführt werden gilt auch für die erhaltene syntaktisch separierte zu F äquivalente Formel R wieder $mk(R) = mk(F)$.

■

Satz 100 *Zu jeder LTL-Formel F existiert eine syntaktisch separierte Formel R, die über der Zeitstruktur der ganzen Zahlen zu F äquivalent ist.*

Beweis:
Nach den obigen Vorbereitungen genügt eine einfache Induktion über den Formelaufbau. Sind A,B syntaktisch separierte Formeln, so trivialerweise auch $\neg A$, $A \wedge B$ und $A \vee B$ und wegen Lemma 99 auch $(A\ since\ B)$ und $(A\ until\ B)$.

■

4.1.4 Übungsaufgaben

Übungsaufgabe 4.1.1 *In den Voraussetzungen zu Satz 93 wird verlangt, daß die Strukturen in $\mathcal{M}$ linear sind. Wo wird diese Voraussetzung im Beweis gebraucht?*

Übungsaufgabe 4.1.2 *Wir betrachten noch einmal die Behauptungen von Lemma 87. Zeigen Sie*

1. *Die Behauptungen 1. und 2. gelten für beliebige Zeitstrukturen.*
2. *Geben Sie Gegenbeispiele, die zeigen, daß 3. und 4. für nicht lineare Zeitstrukturen falsch werden können.*

Übungsaufgabe 4.1.3 *Finden Sie eine LTL-Formel A_{alt}, so daß für alle temporale Strukturen $\mathcal{K}$ mit Zeitstruktur $(\mathbb{N}, <)$ gilt*

$$\begin{gathered}(\mathcal{K}, 0) \models A_{alt} \\ \textit{gdw} \\ (\mathcal{K}, i) \models p \Leftrightarrow i \textit{ ist gerade.}\end{gathered}$$

Beachten Sie den Unterschied zu nicht definierbaren Eigenschaft in Lemma 88.

4.2 Verzweigende Temporale Logiken

4.2.1 Einführung

In diesem Abschnitt betrachten wir als einen typischen Vertreter der temporalen Logiken für verzweigende Zeitstrukturen die *Computation Tree Logic* kurz *CTL*. Wir orientieren uns an der Darstellung in [Clarke *et al.*, 1986]. Eine umfangreiche Übersicht über modale Temporallogiken findet man in [Emerson, 1992]. Eine zusammenfassende Darstellung des Modellprüfungsverfahrens mit allen dazu erforderlichen Vorausetzungen findet man z.B.in [Clarke *et al.*, 2001].
CTL ist eine Aussagenlogik. Die Menge der aussagenlogischen Variablen bezeichnen wir mit P_0.

Definition 110 (CTL-Formeln)

1. *jede aussagenlogische Variable ist eine CTL-Formel,*
2. *Sind F, G CTL-Formeln, dann auch die aussagenlogischen Kombinationen davon, d.h. $\neg F$, $F \vee G$, $F \wedge G$, etc.,*
3. *sind F, G CTL-Formeln, dann auch* $(\mathrm{all\,next}\,F)$, $(\mathrm{ex\,next}\,F)$, $\mathrm{all}(F\,\mathrm{until}\,G)$ *und* $\mathrm{ex}(F\,\mathrm{until}\,G)$.

Die Bedeutung der modalen Operatoren all, ex, next und until wird in Kürze erklärt werden. Es gibt mehr als einen Zugang zur Semantik von CTL. Wir beginnen mit dem abstraktesten, dem Zugang über eine eingeschränkte Klasse von Kripke Strukturen.

Definition 111
Eine Kripke Struktur $\mathcal{K} = (G, R, v)$ heißt eine **Berechnungsstruktur**, *wenn (G, R) eine partiell geordnete Menge ist, in der jedes Element $g \in G$ mindestens einen direkten Nachfolger besitzt.*

Definition 112 (Pfade in Kripke Strukturen)
Sei $\mathcal{K} = (G, R, v)$ eine Berechnuungsstruktur.
Ein **Pfad** *in $\mathcal{K}$ ist eine unendliche Folge $g_0, \ldots, g_n \ldots$ von Welten $g_n \in G$, so daß jedes g_{i+1} ein direkter Nachfolger von g_i ist.*

Definition 113 (Modellbegriff für CTL-Formeln)
Sei $\mathcal{K} = (G, R, v)$ eine Berechnungsstruktur
Der Einfachheit halber schreiben wir $g \models F$ anstelle von $(\mathcal{K}, g) \models F$.

1	$g \models p$	*gdw*	$v(g,p) = 1$
2	$g \models \neg F$	*gdw*	$g \not\models F$
3	$g \models F_1 \wedge F_2$	*gdw*	$g \models F_1$ *und* $g \models F_2$
4	$g \models \text{all next}\, F$	*gdw*	*für alle direkten Nachfolger* g_1 *von* g *gilt* $g_1 \models F$,
5	$g \models \text{ex next}\, F$	*gdw*	*es gibt einen direkten Nachfolger* g_1 *von* g *mit* $g_1 \models F$
6	$g \models \text{all}(F_1 \,\text{until}\, F_2)$	*gdw*	*für jeden Pfad* $g_0, g_1, \ldots$ *mit* $g_0 = g$ *gibt es ein* $i \geq 0$, *so daß* $g_i \models F_2$ *und* $g_j \models F_1$ *für alle* j *mit* $0 \leq j < i$,
7	$g \models \text{ex}(F_1 \,\text{until}\, F_2)$	*gdw*	*es gibt einen Pfad* $n_0, n_1, \ldots$ *mit* $n_0 = n$ *und ein* $i \geq 0$, *so daß* $n_i \models F_2$ *und* $n_j \models F_1$ *für alle* j *mit* $0 \leq j < i$,

Neben den temporalen Basis-Operatoren in CTL werden häufig die folgenden definierten Operatoren benutzt.

Definition 114 (Definierte temporale Operatoren)

$uv(F)$	$\equiv$	$\text{all}(true \,\text{until}\, F)$	*F ist unvermeidlich*
$er(F)$	$\equiv$	$\text{ex}(true \,\text{until}\, F)$	*F ist ereichbar*
$efa(F)$	$\equiv$	$\neg\,\text{all}(true \,\text{until}\, \neg F)$	*ein für alle mal F*
$i(F)$	$\equiv$	$\neg\,\text{ex}(true \,\text{until}\, \neg F)$	*immer F*

Lemma 101 *Sei $\mathcal{K} = (G, R, v)$ eine Berechnungsstruktur und $g \in G$. Wir schreiben wieder $g \models F$ anstelle von $(\mathcal{K}, g) \models F$.*

1	$g \models uv(F)$	*gdw*	*für jeden Pfad* $g_0, g_1, g_2, \ldots$ *mit* $g_0 = g$ *gibt es ein* $i \geq 0$ *mit* $g_i \models F$
2	$g \models er(F)$	*gdw*	*es gibt einen Pfad* $g_0, g_1, g_2, \ldots$ *mit* $g_0 = g$ *und ein* $i \geq 0$ *mit* $g_i \models F$
3	$g \models efa(F)$	*gdw*	*es gibt einen Pfad,* $g_0 g_1, g_2, \ldots$ *mit* $g_0 = g$ *so daß für alle* $i \geq 0$ *gilt* $g_i \models F$
4	$g \models i(F)$	*gdw*	*für jeden Pfad* $g_0, g_1, g_2, \ldots$ *mit* $g_0 = g$ *gilt* $g_i \models F$ *für alle* $i \geq 0$

In der Definition einer Berechnungsstruktur wurde festgelegt, daß die partielle Ordnung (G, R) diskret ist, d.h. gilt $g_0 R g_1$ für $g_0, g_1 \in G$, dann gibt es ein $h \in G$ mit $g_0 R h$, so daß zwischen g_0 und h kein weitere Zeitpunkt mehr liegt. Das macht Sinn für die Modellierung diskreter Systeme, wie sie z.B. in Zustandsübergangssystemen vorliegen. Die Interpretation der temporalen Operatoren, die next enthalten, könnte man zwar auch in nicht diskreten Ordnungen erklären, aber sie wäre unnatürlich und gewöhnungsbedürftig.
Außerdem wurde festgelegt, daß (G, R) keine Endpunkte enthält. Das ist weniger zwingend erforderlich, aber bequem, siehe Lemma 105 und auch Übungsaufgabe 4.2.1.

Ein konkreteres Semantikmodell entsteht, wenn man zusätzlich zu den Forderungen an eine Berechungsstuktur verlangt, daß der darin enthaltene Rahmen (G, R) eine Baumordnung mit eindeutigem Wurzelknoten darstellt. Diese Strukturen werden dann Berechnungsbäume genannt, im Englischen *computation trees*, und haben der Logik ihren Namen gegeben. Die Unterschiede zwischen beiden Modellen sind nicht gravierend, da man jede Berechnungsstruktur in einen Berechnungsbaum auffalten kann. Wir skizzieren kurz, wie man das macht.

Definition 115 *Sei* $\mathcal{K} = (G, R, v)$ *eine Berechnungsstruktur und* $s \in G$.

1. *Eine Berechnungsfolge für* $(\mathcal{K}, s)$ *ist eine endliche Folge von Zuständen (Welten)* $s_i \in G$*, so daß* $s_0 = s$ *und für alle* s_{i+1} *in der Folge ist* s_{i+1} *ein unmittelbarer Nachfolger von* s_i.

2. *Eine Berechnungsfolge P_1 ist ein unmittelbarer Nachfolger einer Berechnungsfolge P_0, wenn $P_1 = \langle P_0, s \rangle$ ist, d.h. P_0 ist ein Anfangsstück von P_1 und P_1 ist genau um ein Element länger als P_0.*
3. *Der Berechnungsbaum $\mathcal{T} = (T, R^*, v)$ zu $(\mathcal{K}, s)$ entsteht, indem wir v unverändert aus der Berechnungsstruktur übernehmen, für T die Menge aller Berechnungsfolgen für $(\mathcal{K}, s)$ einsetzen und R^* als die unmittelbare Nachfolgerrelation auf Berechungsfolgen definiert.*

Satz 102 *In jeder Berechnungsstruktur $\mathcal{K}$ gelten die folgenden Äquivalenzen:*

1. $i(F) \leftrightarrow F \wedge \mathrm{all\,next}\, i(F)$
2. $\mathrm{ex}(F_1 \,\mathrm{until}\, F_2) \leftrightarrow F_2 \vee (F_1 \wedge \mathrm{ex\,next}(F_1 \,\mathrm{until}\, F_2))$

Beweis Einfach.

■

4.2.2 Fixpunkte

Als Vorbereitung auf den im nächsten Unterkapitel behandleten Modellprüfungsalgorithmus stellen wir hier einige allgemeine Resultate über Fixpunkten vor.

Definition 116 *Monotonie*

Definition 117 *Fixpunkte*
Eine Funktion $f : \mathcal{P}(G) \to \mathcal{P}(G)$ von der Menge $\mathcal{P}(G)$ aller Teilmengen einer Menge G in $\mathcal{P}(G)$ und Z eine Teilmenge von G.

1. *Z heißt* Fixpunkt *von f falls $f(Z) = Z$ gilt.*
2. *Z heißt kleinster* Fixpunkt *von f, wenn Z ein Fixpunkt ist und für jeden anderen Fixpunkt U von f gilt $Z \subseteq U$.*
3. *Z heißt größter* Fixpunkt *von f , wenn Z ein Fixpunkt ist und für jeden anderen Fixpunkt U von f gilt $U \subseteq Z$.*

Mit $f^n(M)$ bezeichnen wir, wie üblich, die n-fache Hintereinanderausführung von f, d.h. $f^1(M) = f(M)$, $f^{n+1}(M) = f(f^n(M))$.

Lemma 103
Gegeben sei eine montone Funktion $f : \mathcal{P}(G) \to \mathcal{P}(G)$ auf einer endlichen *Menge G.*

1. *Dann besitzt f besitzt einen kleinsten und einen größten Fixpunkt.*
2. *$\bigcup_{n\geq 1} f^n(\emptyset)$ ist der kleinste Fixpunkt von f.*
3. *$\bigcap_{n\geq 1} f^n(G)$ ist der größte Fixpunkt von f.*

Beweis

Zu 1: Aus der Monotonie folgt zunächst

$$\emptyset \subseteq f(\emptyset) \subseteq f^2(\emptyset) \subseteq \ldots \subseteq f^n(\emptyset) \subseteq \ldots$$

Da G endlich ist, muß ein i existieren mit $f^i(\emptyset) = f^{i+1}(\emptyset)$. Dann ist $Z = \bigcup_{n\geq 1} f^n(\emptyset) = f^i(\emptyset)$ ein Fixpunkt von f,

$$f(Z) = f(f^i(\emptyset)) = f^{i+1}(\emptyset) = f^i(\emptyset) = Z$$

Sei jetzt U ein weiterer Fixpunkt für f. Wegen $\emptyset \subseteq U$ folgt aus der Monotonie von f zunächst $f(\emptyset) \subseteq f(U) = U$. Durch Induktion erhält man für jedes n auch $f^n(\emptyset) \subseteq U$. Also auch $Z = f^i(\emptyset) \subseteq U$.

Zu 2: analog.

∎

Die nachfolgenden Überlegungen sind für die Modellprüfung irrelevant. Wir wollen hier die Frage stellen, ob Lemma 103 auch für unendliche Trägermengen G gilt.

Definition 118 *Stetigkeit*
Eine Funktion $f : \mathcal{P}(G) \to \mathcal{P}(G)$ von der Menge $\mathcal{P}(G)$ aller Teilmengen einer Menge G in $\mathcal{P}(G)$ heißt

1. $\cup$-**stetig**, *wenn für jede aufsteigende Folge $M_1 \subseteq M_2 \subseteq \ldots \subseteq M_n \subseteq \ldots$ gilt*

$$f(\bigcup_{n\geq 1} M_n) = \bigcup_{n\geq 1} f(M_n)$$

2. $\cap$-**stetig**, *wenn für jede absteigende Folge $M_1 \supseteq M_2 \supseteq \ldots \supseteq M_n \supseteq \ldots$ gilt*

$$f(\bigcap_{n\geq 1} M_n) = \bigcap_{n\geq 1} f(M_n)$$

Offensichtlich ist jede stetige Funktion ($\cup$-stetig oder $\cap$-stetig) auch monoton. Dazu braucht man nur die aufsteigende Folge $M = M_1$ und $N = M_n$ für alle $n > 1$ zu betrachten. Die Umkehrung gilt fr unendiche Trägermengen im allgemeinen nicht. Für stetige Funktionen gilt Lemma 103 auch im unendlichen Fall.

Lemma 104
Gegeben sei eine $\cup$-stetige Funktion $f : \mathcal{P}(G) \to \mathcal{P}(G)$ und eine $\cap$-stetige Funktion $g : \mathcal{P}(G) \to \mathcal{P}(G)$. Dann gilt für jedes $M, N \in \mathcal{P}(G)$ mit $M \subseteq f(M)$ und $g(N) \subseteq N$

1. *$\bigcup_{n\geq 1} f^n(M)$ ist der kleinste Fixpunkt von f oberhalb von M,*
2. *$\bigcap_{n\geq 1} g^n(M)$ ist der größte Fixpunkt von g unterhalb von M,*

Beweis:
Zu 1: Aus der Monotonie folgt zunächst

$$M \subseteq f(M) \subseteq f^2(M) \subseteq \ldots \subseteq f^n(M) \subseteq \ldots$$

Setze $P = \bigcup_{n\geq 1} f^n(M)$. Dann gilt sicherlich $M \subseteq P$. Außerdem gilt

$$\begin{array}{rcll} f(P) & = & f(\bigcup_{n\geq 1} f^n(M)) & \\ & = & \bigcup_{n\geq 1} f^{n+1}(M) & \text{wegen Stetigkeit} \\ & = & \bigcup_{n\geq 1} f^n(M) & \text{da } f(M) \subseteq f^2(M) \\ & = & P & \end{array}$$

Sei jetzt Q ein weiterer Fixpunkt für f mit $M \subseteq Q$. Wegen Monotonie und Fixpunkteigenschaft folgt $f(M) \subseteq F(Q) = Q$ und weiter für jedes $n \geq 1$ auch $f^n(M) \subseteq Q$. Also auch $P = \bigcup_{n\geq 1} f^n(M) \subseteq Q$

Zu 2: analog.

∎

Auch monotone, nicht notwendigerweise stetige Mengenfunktionen auf $\mathcal{P}(G)$ besitzen kleinste und größte Fixpunkte, die sich aber nicht unbedingt durch einfache Iteration von f wie in Lemma 103 und 104 angeben lassen.

4.2.3 CTL Modellprüfung - Ein Beispiel

Modellprüfungsalgorithmen (engl. model checking algorithms) sind in der Lage zu einer gegebenen aussagenlogischen Kripke

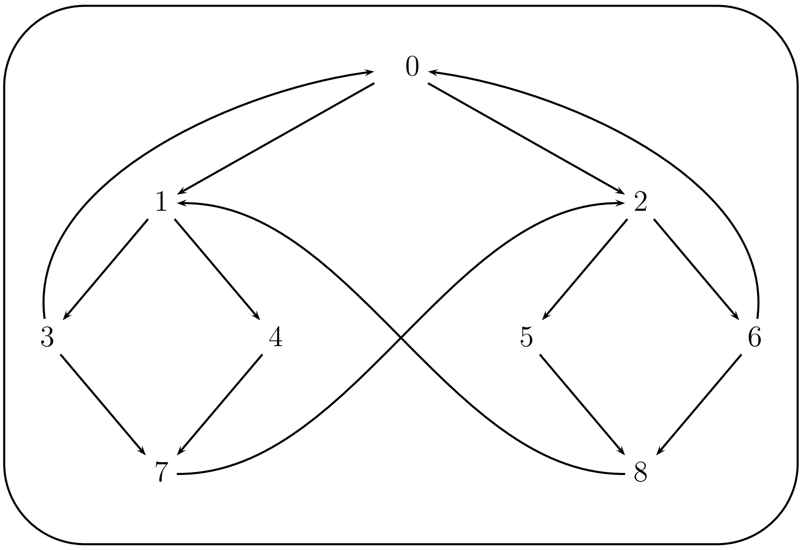


Abbildung 4.3: Kripke Struktur $\mathcal{K}_{me}$ zum Zugriffskontrollproblem

Struktur $\mathcal{K} = (G, R, v)$, einem Zustand $s \in G$ und zu einer gegebenen CTL-Formel A zu entscheiden, ob $(\mathcal{K}, s) \models A$ gilt oder nicht. Erstaunlich viele Aufgabenstellungen in der Verifikation oder Validierung von Schaltkreisen, Protokollen und allgemeiner in der Programmierung verteilter Systeme lassen sich auf dieses Grundmuster zurückführen. Modellprüfungsverfahren haben deshalb auch schon Eingang gefunden in die industrielle Praxis.
Die folgende Darstellung folgt im wesentlichen der Arbeit [Clarke *et al.*, 1986], insbesondere stammt das hier wiederholte Beispiel daraus.

Als Beispiel betrachten wir die Kripke Struktur in Abbildung 4.3, die bei der Modellelierung eines konkurrierenden Zugriffs zweier Prozesse auf eine gemeinsame Ressource entsteht. Es handelt sich dabei um ein Spielbeispiel, dessen einziger Zweck die Illustration des Modellprüfungsverfahren ist. Es ist eine (Über)Simplifikation des *mutual exclusion* Problems. Eine Darstellung dieses Problems und der Lösungsvorschläge dazu findet man z.B. in [Lamport, 1985]. Zur Formulierung des vereinfachten Problems gehören die aussagenlogischen Variablen

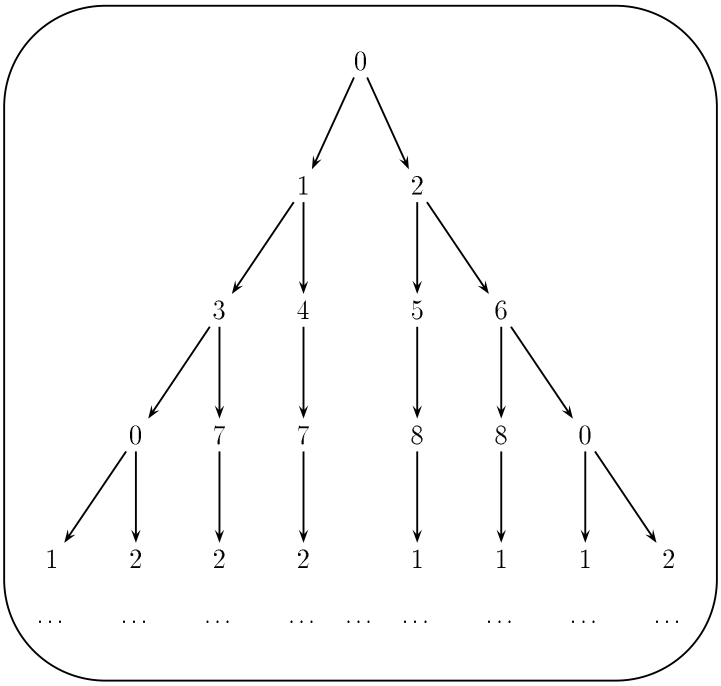


Abbildung 4.4: Berechnungsbaum zu $\mathcal{K}_{me}$

N_i	Prozeß i befindet sich in einer nichtkritischen Region
T_i	Prozeß i befindet sich in der Anmeldephase
C_i	Prozeß i befindet sich in einer kritischen Region

Den Zuständen im Diagramm von Abbildung 4.3 ist dabei die folgende Wahrheitswertbelegung zugeordnet:

	0	1	2	3	4	5	6	7	8
N_1	1	0	1	0	0	0	1	0	0
N_2	1	1	0	1	0	0	0	0	0
T_1	0	1	0	0	1	1	0	0	1
T_2	0	0	1	0	1	1	0	1	0
C_1	0	0	0	1	0	0	0	1	0
C_2	0	0	0	0	0	0	1	0	1

Das Anfangsstück des Berechnungsbaums zu $\mathcal{K}_{me}$ ist in Abbildung 4.4 zu sehen.
Wir illustrieren die Arbeitsweise prinzipielle des model checking Algorithmus anhand eines Beispiels. Wir wollen zeigen, daß in

der Kripke Struktur $\mathcal{K}_{me}$ in dem Zustand 1 die Formel

$$F = T_1 \to uv(C_1) = \neg T_1 \vee \mathrm{all}(true\,\mathrm{until}\,C_1)$$

wahr ist. Dieses Problem wird in drei Hauptschritte untergliedert:

1. Zuerst werden alle Teilformeln der Eingabeformel ermittelt. Im vorliegenden Beispiel sind das

$$\{T_1, C_1, \neg T_1, \mathrm{all}(true\,\mathrm{until}\,C_1), F\}$$

2. Für jede dieser Teilformeln C wird die Menge der Zeitpunkte, $\tau(C)$, zu denen C wahr ist, berechnet:

$$\tau(C) = \{s \mid (\mathcal{K}_{me}, s) \models C\}$$

3. Zum Schluß muß dann noch abgefragt werden, ob $1 \in \tau(F)$ gilt.

Wir müssen noch erklären wie Teil 2 gelöst wird. Zuerst wird $\tau(C)$ für die einfachsten Teilformeln bestimmt, d.h. für die Aussagenvariablen. Offensichtlich kann $\tau(C)$ direkt aus der Kripke Struktur abgelesen werden, $\tau(C) = \{s \in G \mid v(s,C) = 1\}$. In unserem Beispiel ergibt sich:

$$\begin{array}{lcl} \tau(T_1) & = & \{1,4,5,8\} \\ \tau(C_1) & = & \{3,7\} \end{array}$$

Ist F_0 eine Teilformel, die aus F_1 und F_2 mittels aussagenlogischer Operatoren aufgebaut ist, so kann $\tau(F_0)$ durch die entsprechenden Mengenoperationen aus $\tau(F_1)$ und $\tau(F_2)$ berechnet werden:

$$\begin{array}{lcl} \tau(\neg F_1) & = & G \setminus \tau(F_1) \\ \tau(F_1 \wedge F_2) & = & \tau(F_1) \cap \tau(F_2) \\ \tau(F_1 \vee F_2) & = & \tau(F_1) \cup \tau(F_2) \end{array}$$

Für unsere Beispielrechnung erhalten wir zunächst nur

$$\tau(\neg T_1) \quad = \quad \{0,2,3,6,7\}$$

die Auswertung der Disjunktion muß noch warten bis das zweite Disjunktionsglied ausgewertet ist. Interessanter ist die Berechnung von $\tau(F_0)$ für Formeln F_0, die mittels temporaler Operatoren aufgebaut sind, etwa $F_0 = \mathrm{all}(F_1 \,\mathrm{until}\, F_2)$. In diesen Fällen wird $\tau(F_0)$ durch eine Fixpunktberechnung gefunden. Dabei sind sowohl kleinste Fixpunkte als auch größte Fixpunkte von Interesse.

Soll $\tau(F_0) = \tau(\mathrm{all}(F_1 \,\mathrm{until}\, F_2))$ berechnet werden, so beginnt man mit $\tau^0(F_0) = \emptyset$. $\tau^1(F_0)$ enthält dann alle Zeitpunkte, für die $\mathrm{all}(F_1 \,\mathrm{until}\, F_2)$ aus dem einfachen Grund gilt, daß schon F_2 wahr ist, d.h. $\tau^1(F_0) = \tau(F_2)$. Im nächsten Schritt enthält $\tau^2(F_0)$ nach wie vor die Menge der Zeitpunkte $\tau(F_2)$, aber jetzt auch noch die Zeitpunkte, zu denen F_1 gilt und in allen nächsten Schritten F_2 wahr ist:

$$\tau^2(F_0) = \tau(F_2) \cup \{s \in \tau(F_1) \mid \text{für alle } t \text{ mit } sRt \text{ gilt } t \in \tau^1(F_0)\}$$

Im allgemeinen

$$\tau^{n+1}(F_0) = \tau(F_2) \cup \{s \in \tau(F_1) \mid \text{für alle } t \text{ mit } sRt \text{ gilt } t \in \tau^n(F_0)\}$$

Diese Berechnung wird solange iteriert bis $\tau^{n+1}(F_0) = \tau^n(F_0)$ gilt. Dann setzt man $\tau(F_0) = \tau^n(F_0)$. In unserem Beispiel ergibt sich

$$\begin{array}{lcl}
\tau^0(\mathrm{all}(\mathit{true} \,\mathrm{until}\, C_1)) & = & \emptyset \\
\tau^1(\mathrm{all}(\mathit{true} \,\mathrm{until}\, C_1)) & = & \{3,7\} \\
\tau^2(\mathrm{all}(\mathit{true} \,\mathrm{until}\, C_1)) & = & \{3,4,7\} \\
\tau^3(\mathrm{all}(\mathit{true} \,\mathrm{until}\, C_1)) & = & \{1,3,4,7\} \\
\tau^4(\mathrm{all}(\mathit{true} \,\mathrm{until}\, C_1)) & = & \{1,3,4,7,8\} \\
\tau^5(\mathrm{all}(\mathit{true} \,\mathrm{until}\, C_1)) & = & \{1,3,4,5,7,8\} \\
\tau^6(\mathrm{all}(\mathit{true} \,\mathrm{until}\, C_1)) & = & \{1,3,4,5,7,8\}
\end{array}$$

Somit erhalten wir $\tau(\mathrm{all}(\mathit{true} \,\mathrm{until}\, C_1)) = \{1,3,4,5,7,8\}$ und

$$\begin{array}{lcl}
\tau(F) & = & \tau(\neg T_1) \cup \tau(\mathrm{all}(\mathit{true} \,\mathrm{until}\, C_1)) \\
 & = & \{0,2,3,6,7\} \cup \{1,3,4,5,7,8\} \\
 & = & G
\end{array}$$

Die Mengenfunktion, deren Fixpunkt durch die Approximationen

τ^n berechnet wird, war bisher nur implizit aufgetreten. Explizit ist diese Funktion $f_{F_0}(Z)$ definiert durch

$$f_{F_0}(Z) = \tau(F_2) \cup (\tau(F_1) \cap f_{AX}(Z))$$

wobei $f_{AX}(Z)) = \{s \mid \text{für alle } t \text{ mit } sRt \text{ gilt } t \in Z\}$

4.2.4 CTL Modellprüfung - Theorie

Wir wollen die Zuordnung von CTL-Formeln F zu Mengenfunktionen $f_F(Z)$ jetzt systematisch betrachten.

Definition 119 (Mengenfunktionen für CTL-Formeln)
Sei $\mathcal{K} = (G, R, v)$ eine Kripke Struktur, F eine CTL-Formel, dann werden die Mengenfunktionen $f_F(Z) : \mathcal{P}(G) \to \mathcal{P}(G)$ wir folgt definiert:

$$\begin{array}{lll}
F = \mathrm{all}(F_1 \,\mathrm{until}\, F_2) & f_F(Z) = & \tau(F_2) \cup (\tau(F_1) \cap f_{AX}(Z)) \\
F = \mathrm{ex}(F_1 \,\mathrm{until}\, F_2) & f_F(Z) = & \tau(F_2) \cup (\tau(F_1) \cap f_{EX}(Z)) \\
F = uv(F) & f_F(Z) = & \tau(F) \cup f_{AX}(Z) \\
F = er(F) & f_F(Z) = & \tau(F) \cup f_{EX}(Z) \\
F = i(F) & f_F(Z) = & \tau(F) \cap f_{AX}(Z) \\
F = efa(F) & f_F(Z) = & \tau(F) \cap f_{EX}(Z)
\end{array}$$

wobei

$$\begin{array}{lll}
f_{AX}(Z) & = & \{s \in G \mid \textit{für alle } t \textit{ mit } sRt \textit{ gilt } t \in Z\} \\
f_{EX}(Z) & = & \{s \in G \mid \textit{es gibt ein } t \textit{ mit } sRt \textit{ und } t \in Z\} \\
\textit{und} & & \\
\tau(F) & = & \{g \in G \mid (\mathcal{K}, g) \models F\}
\end{array}$$

Lemma 105 *Sei $\mathcal{K} = (G, R, v)$ eine Kripke Struktur und F eine CTL-Formel. Für dieses Lemma ist vorausgesetzt, da/ss $\mathcal{K}$ keine Endpunkte besitzt, d.h. zu jedem $g \in G$ gibt es ein h mit gRh. Dann gilt*

1. *für $F = \mathrm{all}(F_1 \,\mathrm{until}\, F_2)$,$\mathrm{ex}(F_1 \,\mathrm{until}\, F_2)$, $uv(F)$, $er(F)$*

$$\tau(F) = kfp\ Z\ f_F(Z)$$

2. *für* $F = i(F)$, $efa(F)$

$$\tau(F) = gfp\ Z\ f_F(Z)$$

wobei $kfp\ Z\ f(Z)$ *für den kleinsten und* $gfp\ Z\ f(Z)$ *für den größten Fixpunkt der Funktion* $f(Z)$ *steht.*

Beweis: Einige Fälle des Beweises werden in [Clarke *et al.*, 1993, Clarke *et al.*, 2001] vorgerechnet.

Der Fall $i(F)$

Als erstes bemerken wir, daß für jede Berechnungsstruktur $\mathcal{K} = (G, R, v)$ und jeden Zustand $g \in G$ die folgende Äquivalenz gilt:

$$g \models i(F) \Leftrightarrow g \models F \text{ und für alle } h \text{ mit } gRh \text{ gilt } h \models i(F)$$

In der τ-Notation liest sich das als

$$\tau(i(F)) = \tau(F) \cap \{g \mid \text{ für alle } h \text{ mit } gRh \text{ gilt } h \in \tau(i(F))\}$$

Vergleicht man das mit den Definitionen in Definition 119 so sieht man, daß sich das auch schreiben läßt als

$$\tau(i(F)) = f_{i(F)}(\tau(i(F)))$$

Damit ist $\tau(i(F))$ als ein Fixpunkt der Funktion $f_{i(F)}$ erkannt.

Bleibt noch zu zeigen, daß es der grösste Fixpunkt von $f_{i(F)}$ ist. Sei dazu H irgendein Fixpunkt von $f_{i(F)}$, d.h. $f_{i(F)}(H) = H$. Wir wollen $H \subseteq \tau(i(F))$ zeigen. Sei also $g_0 \in H$. Wir müssen nachweisen, daß

$$\text{für alle } n \geq 0 \text{ und alle } g_i \text{ mit } g_{i-1}Rg_i \text{ für alle } 1 \leq i \leq n \text{ gilt } g_n \in \tau(F)$$

gilt, denn das ist gerade die Definition von $g_0 \in \tau(i(F))$.

Wir zeigen diese Behauptung indem wir zwei Dinge beweisen

1. $H \subseteq \tau(F)$
2. für alle $n \geq 0$ und alle g_i mit $g_{i-1}Rg_i$ für alle $1 \leq i \leq n$ gilt $g_n \in H$

zu 1: $H = f_{i(F)}(H) = \tau(F) \cap f_{AX}(H) \subseteq \tau(F)$
zu 2: Das beweisen wir durch Induktion nach n. Für $n = 0$ ist nichts zu zeigen. Im Schritt von $n - 1$ auf n haben wir als Induktionsvoraussetzung $g_{n-1} \in H$ und $g_{n-1} R g_n$. Wegen der vorausgesetzten Fixpunkteigenschaft von H gilt

$$g_{n-1} \in H = f_{i(F)}(H) = \tau(F) \cap f_{AX}(H)$$

Also insbesondere

$$g_{n-1} \in f_{AX}(H) = \{g \mid \text{ für alle } h \text{ mit } gRh \text{ gilt } h \in H\}$$

Daraus folgt unmittelbar $g_n \in H$ und wir sind fertig.

Der Fall $F = \mathrm{ex}(F_1 \,\mathrm{until}\, F_2)$

Als erstes bemerken wir, daß für jede Berechnungsstruktur $\mathcal{K} = (G, R, v)$ und jeden Zustand $g \in G$ die folgende Äquivalenz gilt:

$$\begin{array}{lcl} g \models \mathrm{ex}(F_1 \,\mathrm{until}\, F_2) & \Leftrightarrow & g \models F_2 \text{ oder } (g \models F_1 \text{ und} \\ & & \text{es gibt } h \text{ mit } gRh \text{ und } h \models \mathrm{ex}(F_1 \,\mathrm{until}\, F_2)) \end{array}$$

Umgeschrieben ergibt sich daraus:

$$\begin{array}{lcl} \tau(\mathrm{ex}(F_1 \,\mathrm{until}\, F_2)) & = & \tau(F_2) \cup (\tau(F_1) \cap \\ & & \{g \mid \text{ es gibt } h \text{ mit } gRh \text{ und } h \in \tau(\mathrm{ex}(F_1 \,\mathrm{until}\, F_2))\}) \end{array}$$

Nach Definition von f_{EX}:

$$\tau(\mathrm{ex}(F_1 \,\mathrm{until}\, F_2)) = \tau(F_2) \cup (\tau(F_1) \cap f_{EX}(\tau(\mathrm{ex}(F_1 \,\mathrm{until}\, F_2))))$$

Nach Definition von f_F

$$\tau(\mathrm{ex}(F_1 \,\mathrm{until}\, F_2)) = f_F(\tau(\mathrm{ex}(F_1 \,\mathrm{until}\, F_2)))$$

$\tau(\mathrm{ex}(F_1 \,\mathrm{until}\, F_2))$ ist also ein Fixpunkt von f_F. Es bleibt zu zeigen, daß es der keinste Fixpunkt von f_F ist. Sei also H ein weitere Fixpunkt, $f_F(H) = H$. Sei $g_0 \in \tau(\mathrm{ex}(F_1 \,\mathrm{until}\, F_2))$. Nach der Semantik von $\mathrm{ex}(F_1 \,\mathrm{until}\, F_2)$ gibt es $n \in \mathbb{N}$ und g_i für $1 \leq i \leq n$ mit

1. $g_i R g_{i+1}$ für alle $0 \leq i < n$.
2. $g_n \in \tau(F_2)$.
3. $g_i \in \tau(F_1)$ für alle $0 \leq i < n$.

Wir zeigen durch Induktion nach n daß $g_0 \in H$ gilt.

$\mathbf{n} = 0$ Wegen der Fixpunkteigenschaft von H gilt

$$H = f_F(H) = \tau(F_2) \cup (\tau(F_1) \cap f_{EX}(H)) \supseteq \tau(F_2)$$

$\mathbf{n}-1 \leadsto n$ Als Induktionsvoraussetzung haben wir $g_1 \in H$. Wegen $g_0 \in \tau(F_1)$ und $g_0 R g_1$ haben wir $g_0 \in (\tau(F_2) \cap f_{EX}(H)) \subseteq \tau(F_2)$ und damit auch $g_0 \in H$.

Der Fall $F = \mathrm{all}(F_1 \,\mathrm{until}\, F_2)$

Als ersten Schritt zeigen wir wieder, daß $\tau(F)$ ein Fixpunkt der in Definition 119 definierten Funktion f_F ist.Wir begnügen uns jetzt mit der Angabe der grundlegenden Äquivalenz, den weiteren Verlauf des Arguments hat der Leser ja schon zweimal gesehen:

$$\begin{array}{lcl} g \models \mathrm{all}(F_1 \,\mathrm{until}\, F_2) & \Leftrightarrow & g \models F_2 \text{ oder } (g \models F_1 \text{ und} \\ & & \text{für alle } h \text{ mit } gRh \text{ gilt } h \models \mathrm{all}(F_1 \,\mathrm{until}\, F_2)) \end{array}$$

Es bleibt zu zeigen, daß $\tau(F)$ der kleinste Fixpunkt von f_F ist. Sei also H ein weiterer Fixpunkt, $H = f_F(H)$. Wir betrachten ein $g_0 \in \tau(F)$ mit dem Ziel $g_0 \in H$ zu zeigen. Aus $g_0 \in \tau(F)$ folgt mit der in Definition 113 festgelegten Semantik, daß für jeden in g_0 beginnenden Pfad $P_i \equiv g_0 = g_0^i, g_1^i, \ldots, g_j^i, \ldots$ der Berechnungsstruktur $\mathcal{K}$ ein n_i existiert mit

1. $g_{n_i}^i \models F_2$
2. für alle j mit $0 \leq j < n_i$ gilt $g_j^i \models F_1$

Wieviele solcher Pfade gibt es denn? Man könnte vielleicht befürchten, daß es unendlich viele davon geben könnte. Die folgende Überlegung zeigt, daß das nicht sein kann. Wir müssen von dem Pfad P_i nur das endliche Anfangsstück bis $g_{n_i}^i$ betrachten. Hätten wir unendlich viele endliche Pfade, so hätten wir unendlich

viele Knoten. Königs Lemma führt nun zu einem Widerspruch, es besagt nämlich, daß jeder unendliche, endlich-verzweigende Baum einen unendlichen Pfad haben muß. Wir wissen jetzt also, daß wir nur endlich viele in g_0 beginnende Pfade betrachten müssen, sagen wir es gibt davon k viele. Mit n bezeichnen wir das Maximum von $\{n_i \mid 0 \leq j \leq k\}$.
Wir zeigen $g_0 \in H$ durch Induktion nach n.

n $= 0$ In diesem Fall gilt $g_0 \in \tau(F_2)$. Es gilt außderdem $H = f_F(H) = \tau(F_2) \cup (\tau(F_1 \cap f_{AX}(H)) \supseteq \tau(F_2)$. Damit haben wir in diesem Fall $g_0 \in H$ gezeigt.

$(n-1) \rightsquigarrow n$ Nach Induktionsvoraussetzung gilt für alle $0 \leq j \leq k$ schon $g_1^j \in H$. Außerdem wissen wir $g_0 \in \tau(F_1)$ und $\{g_1^j \mid 0 \leq j \leq k\} = \{h \mid g_0Rh\}$. Alles zusammengenommen folgt daraus $g_0 \in (\tau(F_1 \cap f_{AX}(H))$ und damit $g_0 \in H$.

∎

Zur Berechnung von $\tau(F)$ genügt es also den kleinsten (resp. größten) Fixunkt von f_F zu berechnen, was mit der Methode aus dem Beweis von Lemma 103 geschehen kann.

Satz 106 (Komplexität)

1. *Das CTL Modellprüfungsproblem ist linear.*
   Genauer gesagt, gibt es einen Algorithmus der für jede CTL-Formel F mit m Zeichen und jede Kripke Struktur $\mathcal{K} = (G, R, v))$ und $n = card(G) + card(R)$ in $m \times n$ Schritten berechnet, ob $\mathcal{K} \models F$ gilt oder nicht.
2. *Das CTL* Modellprüfungsproblem ist PSPACE-vollständig.*

Beweis:
1. wurde zuerst in [Clarke *et al.*, 1986] (Theorem 3.1) angegeben. Das Resultat läßt sich auch durch eine sorgfältige Analyse des oben skizzierten Verfahrens bestätigen.
1. wurde ebenfalls zuerst in [Clarke *et al.*, 1986] (Theorem 6.1) gezeigt unter Ausnutzung eines Resultats aus [Clarke & Sistla, 1985].

∎

Satz 107

1. *Das Erfüllbarkeitsproblem für LTL ist PSPACE-vollständig.*
2. *Modellprüfung für LTL ist PSPACE-vollständig.*
3. *Das Erfüllbarkeitsproblem für die Logik $LTL(sif)$, in der nur der temporale Operator sif, aber beliebige aussagenlogische Operatoren, benutzt werden dürfen, ist NP-vollständig.*
4. *Modellprüfung für $LTL(sif)$ ist NP-vollständig.*

Beweis: siehe [Clarke & Sistla, 1985] Theorem 4.1 für 1. und 2. und Theorem 3.5 für 3. und 4.

Entscheidbarkeit der monadischen Logik zweiter Stufe (oder des universellen Fragments der monadischen Logik zweiter Stufe) für verschiedene Ordnungen oder Klassen von Ordnungen wird in Kapitel 15 von [Gabbay *et al.*, 1994] ausführlich behandelt.

4.2.5 Übungsaufgaben

Übungsaufgabe 4.2.1 *Sei $\mathcal{K} = (G, R, v)$ eine Kripke Stuktur, die alle Anforderungen einer Berechnungsstuktur erfüllt mit Ausnahme der Forderung, daß keine Endpunkt existieren dürfen. Ein Zeitpunkt $g \in G$ ist dabei ein Endpunkt, wenn es kein $h \in G$ mit gRh gibt.*
Überlegen Sie sich, daß die folgenden Aussagen äquivalent sind

g *ist ein Endpunkt*
$(\mathcal{K}, g) \models \mathrm{all\,next}\, false$
$(\mathcal{K}, g) \models \neg\, \mathrm{ex\,next}\, true$

Übungsaufgabe 4.2.2 *In [Emerson, 1992] wird eine andere Definition von CTL gegeben, indem zuerst eine allgemeinere Sprache CTL^* definiert wird und CTL als syntaktische Einschränkung von CTL^* erscheint.*

***Definition 120** (CTL^*)*
Die Sprache CTL^* besteht aus zwei Typen von Formeln, den

Zustandsformeln *und den* **Pfadformeln**, *die simultan rekursiv definiert werden durch:*

1. *jede aussagenlogische Variable ist eine Zustandsformel,*
2. *sind F, G Zustandsformeln, dann auch die aussagenlogischen Kombinationen davon, i.e. $\neg F$, $F \vee G$, $F \wedge G$, etc.,*
3. *ist F eine Pfadformel, dann sind $(\mathrm{all}\, F)$, $(\mathrm{ex}\, F)$ Zustandsformeln,*
4. *jede Zustandsformel ist eine Pfadformel,*
5. *sind F, G Pfadformeln, dann auch $\neg F$, $F \vee G$, $F \wedge G$,*
6. *sind F, G Pfadformeln, dann auch $\mathrm{next}\, F$ und $F\, \mathrm{until}\, G$.*

Die Sprache CTL^* ist sehr ausdruckstark. Eine eingeschränkte Version wird erhalten, indem man Boolesche Kombinationen von Pfadformeln verbietet und auch keine Schachtelung der Operatoren next *und* until *zuläßt.*

Definition 121 (Eingeschränkte** CTL^***)

1. *jede aussagenlogische Variable ist eine Zustandsformel,*
2. *sind F, G Zustandsformeln, dann auch die aussagenlogischen Kombinationen davon, i.e. $\neg F$, $F \vee G$, $F \wedge G$, etc.,*
3. *ist F eine Pfadformel, dann sind $(\mathrm{all}\, F)$, $(\mathrm{ex}\, F)$ Zustandsformeln,*
4. *sind F, G Zustandsformeln, dann sind $(\mathrm{next}\, F)$ und $(F\, \mathrm{until}\, G)$ Pfadformeln.*

Zeigen Sie: CTL ist genau die Menge der Zustandsformeln in der eingeschränkten CTL^*-Sprache.

Übungsaufgabe 4.2.3 *Gegeben sei eine montone Funktion $f : \mathcal{P}(G) \to \mathcal{P}(G)$ auf einer* endlichen *Menge G.*

1. *Finden Sie ein Gegenbeispiel zur folgenden Aussage*

   Zu jeder Teilmenge M von G gibt es einen kleinsten Fixpunkt Z von f mit der Eigenschaft $M \subseteq Z$

2. *Finden Sie eine Bedingung an M, unter der diese Aussage richtig wird.*

Übungsaufgabe 4.2.4 *Sei* $F_0 = \mathrm{all}(F_1 \,\mathrm{until}\, F_2)$ *und*

$$f = f_{F_0} = \tau(F_2) \cup (\tau(F_1 \cap f_{AX}(Z)))$$

Zeigen Sie, daß für alle $n \geq 0$ *gilt*

$$f^n(\emptyset) \subseteq f^{n+1}(\emptyset)$$

Kapitel 5

Lösungen

In diesem Kapitel geben wir Lösungen oder Lösungshinweise für einige der Übungsaufgaben.

Zu 2.1 Einführung in die mehrwertige Logik

zu Aufgabe 2.1.3

1.

2. Wir finden sogar ein Gegenbeispiel, wenn t ein Konstantensymbol c ist.

   Sei $\mathcal{M}$ eine dreiwertigen Struktur, deren Universum aus dem einzigen Elment m besteht. Dann muß natürlich $c^{\mathcal{M}} = m$ gelten. Außerdem soll $v_{\mathcal{M}}(q(c)) = u$ gelten. Dann gilt auch $v_{\mathcal{M}}(\forall x q(x)) = u$ und damit auch $v_{\mathcal{M}}(\forall x q(x) \to q(c)) = u \neq 1$.

3.

zu Aufgabe 2.1.4 Ist $A \supset B$ eine Tautologie, dann nach Definition auch $\sim A \vee B$ und damit umso mehr $\sim A \vee \sim C \vee B$. Da $\sim C \vee C$ eine Tautologie ist, trifft dasselbe auch für $\sim A \vee \sim C \vee C$ zu. Zusammengenommen ist auch

$$(\sim A \vee \sim C \vee B) \wedge (\sim A \vee \sim C \vee C)$$

eine Tautologie. Mit Hilfe des Distributivgesetzes erhält man $(\sim A \vee \sim C) \vee (B \wedge C)$. Mit dem deMorganschen Gesetz $\sim (A \wedge C) \vee (B \wedge C)$ i.e. $A \wedge C \supset B \wedge C$.

zu Aufgabe 2.1.5 1, 2 und 3 sind richtig, 4 ist falsch.

zu Aufgabe 2.1.6

1. Das Deduktionstheorem gilt für die schwache Implikation.

   Die Implikation von rechts nach links folgt unmittelbar aus Definition 12 und der Beobachtung, daß für jede dreiwertige

Struktur $\mathcal{M}$ mit $v_{\mathcal{M}}(A \supset B) = 1$ und $v_{\mathcal{M}}(A) = 1$ auch $v_{\mathcal{M}}(B) = 1$ folgt.

Für den Beweis der umgekehrten Richtung ist es am einfachsten anzunehmen, daß für eine dreiwertige Struktur $\mathcal{M}$ $v_{\mathcal{M}}(A \supset B) \neq 1$ gilt. Die wichtige Beobachtung ist nun, daß das nur der Fall sein kann, wenn $v_{\mathcal{M}}(A) = 1$ und $v_{\mathcal{M}}(B) \neq 1$ gilt. Das widerspricht aber $A \vdash_3 B$.

2. Das Deduktionstheorem gilt nicht für die starke Implikation.

   Die Implikation von rechts nach links läßt sich wie in Teil 1 beweisen, da auch aus $v_{\mathcal{M}}(A \to B) = 1$ und $v_{\mathcal{M}}(A) = 1$ wieder $v_{\mathcal{M}}(B) = 1$ folgt.

   Die umgekehrte Richtung ist nicht richtig, denn es gilt sicherlich $A \vdash_3 A$ aber $A \to A$ ist keine 3-wertige Tautologie.

zu Aufgabe 2.1.7

1. einfaches Nachrechnen.
2. Es gilt $u \supset u = 1$ Aber für $\langle 1, 1\rangle$ aus $U(u, u) = \{0, 1\}^2$ gilt $1 \supset 1 = 1$ und für $\langle 1, 1\rangle \in U(u, u)$ gilt $1 \supset 0 = 0$.
3. offensichtlich.
4. Wir geben eine etwas kompliziertere Überlagerung von Funktionen an. Sei $g_1(A) = A$, $g_2(A) = \neg A$ und $f(A, B) = A \vee B$. Es gilt $[f(g_1(A), g_2(A))]^* \neq f^*(g_1^*(A), g_2^*(A)))$

zu Aufgabe 2.1.8

1. Sei $g = f^*$ für ein $f : \{0, 1\}^n \to \{0, 1\}$. Wir zeigen, daß g monoton bzgl. $\leq_i$ ist. Gilt $g(\vec{w}) = u$, dann gilt trivialerweise für alle $\vec{v}$ mit $\vec{w} <_i \vec{v}$ auch $g(\vec{w}) \leq_i g(\vec{v})$. Im folgenden sei $g(\vec{w}) \neq u$

   Als nächstes bemerken wir, daß für alle $\vec{w} \in \{0, u, 1\}^n$ gilt
   $$U(\vec{w}) = \{\vec{v} \in \{0, 1\}^n \mid \vec{w} <_i \vec{v}\}$$
   Aus $\vec{w} <_i \vec{v}$ für $\vec{v} \in \{0, 1\}^n$ folgt somit $f^*(\vec{w}) = f(\vec{v})$ und damit $g(\vec{w}) = g(\vec{v})$. Für $\vec{v}$ mit $\vec{w} <_i \vec{v}$, die nicht notwendigerweise in $\{0, 1\}^n$ liegen gilt $U(\vec{v}) \subseteq U(\vec{w})$ und damit ebenfalls $g(\vec{w}) = g(\vec{v})$.

2. Die Notwendigkeit dieser Bedingung ist klar. Ist g eine maximal monotone Funktion, dann bemerken wir zunchst, daß für $\vec{v} \in \{0,1\}^n$ auch $g(\vec{v}) \in \{0,1\}$ gelten muß: denn in diesem Fall ist $\vec{v}$ ein maximales Element der Ordnung $\leq_i$. Vergrößert man eine monotone Funktion an einer maximalen Argumentstelle, so ist die entstehende Funktion wieder monoton. Wegen der maximalen Monotonie von g kann also $g(\vec{v}) = u$ nicht gelten. Setzen wir also $f = g \downarrow \{0,1\}^n$ dann gilt $f : \{0,1\}^n \to \{0,1\}$. Sei jetzt $\vec{w} \in \{0,u,1\}^n$ beliebig gewählt. Wir wollen $f^*(\vec{w}) = g(\vec{w})$ zeigen. Zunächst beobachten wir, daß gilt

   $$U(\vec{w}) = \{\vec{v} \in \{0,1\}^n \mid \vec{w} \leq_i \vec{v}\}$$

   Nach 2 gilt damit für alle $\vec{v} \in U(\vec{w})$

   $$g(\vec{w}) \leq_i g(\vec{v}) = f(\vec{v})$$

   Wenn also $g(\vec{w}) \in \{0,1\}$ gilt, dann gilt für alle $\vec{v} \in U(\vec{w})$ schon $g(\vec{w}) = g(\vec{v}) = f(\vec{v})$.
   Gilt $g(\vec{w}) = u$ dann behaupten wir, muß es $\vec{u_1}, \vec{u_2}$ mit $\vec{w} <_i \vec{u_1}$ und $\vec{w} <_i \vec{u_2}$ geben, so daß $g(\vec{u_1}) \neq g(\vec{u_2})$ gilt. Ist das nicht der Fall und gilt etwa für alle $\vec{u}$ mit $\vec{w} <_i \vec{u}$ $g(\vec{u}) = a$, dann definiert man

   $$g_1(\vec{v}) = \begin{cases} g(\vec{v}) & \text{falls } \vec{v} \neq \vec{w} \\ a & \text{falls } \vec{v} = \vec{w} \end{cases}$$

   Offensichtlich gilt $g <_i g_1$ und g_1 ist immer noch monoton bzgl. $\leq_i$.

zu Aufgabe 2.1.9 Der Beweis ist eine leichte Umformulierung des Beweises zu Lemma 3.
Der Nachweis geschieht durch Induktion über den Aufbau der Formel A. Ist A eine aussagenlogische Variable, dann ist die induzierte Funktion $f(A)$ die einstellige Identitätsfunktion und die u-0-Kongruenz trivial. Sei $A = A_1 \wedge A_2$ und vorausgesetzt, daß die u-0-Kongruenz von $f(A_1)$ und $f(A_2)$ schon bekannt sei. Für alle

$\vec{a}, \vec{b} \in \{0, u, 1\}^n$ mit $\vec{a} \sim \vec{b}$ gilt dann nach der Wahrheitstabelle für $\wedge$

$$\begin{array}{lll} f(A)(\vec{a}) = 1 & \text{gdw} & f(A_1)(\vec{a}) = 1 \text{ und } f(A_2)(\vec{a}) = 1 \\ & \text{gdw} & f(A_1)(\vec{b}) = 1 \text{ und } f(A_2)(\vec{b}) = 1 \quad \text{nach Ind. Voraus.} \\ & \text{gdw} & f(A)(\vec{b}) = 1 \end{array}$$

Die beiden noch zu betrachtenden Fälle $A = A_1 \vee A_2$ und $A = \sim A$ werden analog gezeigt.

zu Aufgabe 2.1.10 Wir zeigen $v_{\mathcal{N},\beta\circ F}(A) \leq_i v_{\mathcal{M},\beta}(A)$ durch Induktion nach der Komplexität von A.

Sei zunächst A eine atomare Formel $p(t_1, \ldots, t_k)$. Nach Punkt 1 der Definition von $\mathcal{M} \sqsubseteq^F \mathcal{N}$ gilt $I(\mathcal{N}, \beta \circ F)(t_i) = F(I(\mathcal{M}, \beta)(t_i))$. Aus Punkt 2 derselben Definition folgt dann, indem man $b_i = I(\mathcal{M}, \beta)(t_i)$ einsetzt $v_{\mathcal{N},\beta\circ F}(p(t_1, \ldots, t_k)) \leq_i v_{\mathcal{M},\beta}(p(t_1, \ldots, t_k))$.

Sei jetzt A eine aussagenlogische Kombination. Wir führen nur den Fall vor, daß $A = A_1 \circ A_2$ eine aussagenlogische Kombination von zwei Teilformeln ist vor. Es ist aber klar, daß dasselbe Argument auch für einstellige und mehr als zweistellige aussagenlogische Opteratoren gültig ist. Wir beginnen mit

$$v_{\mathcal{N},\beta\circ F}(A_1 \circ A_2) = v_{\mathcal{N},\beta\circ F}(A_1) \circ v_{\mathcal{N},\beta\circ F}(A_2)$$

Nach Induktionsvoraussetzung gilt

$$v_{\mathcal{N},\beta\circ F}(A_i) \leq_i v_{\mathcal{M},\beta}(A_i)$$

für $i \in \{1, 2\}$. Wegen der Monotonie der Standardfortsetzung folgt daraus auch

$$v_{\mathcal{N},\beta\circ F}(A_1) \circ v_{\mathcal{N},\beta\circ F}(A_2) \leq_i v_{\mathcal{M},\beta}(A_1) \circ v_{\mathcal{M},\beta}(A_2)$$

Die Gleichung $v_{\mathcal{M},\beta}(A_1) \circ v_{\mathcal{M},\beta}(A_2) = v_{\mathcal{M},\beta}(A_1 \circ A_2)$ beendet das Argument.

Wir kommen zu dem Fall $A = \exists x A_0$.
Aus $v_{\mathcal{N},\beta\circ F}(A) = 1$ folgt die Existenz eines Elements $b \in N$ mit $v_{\mathcal{N},(\beta\circ F)^b_x}(A_0) = 1$. Wegen der Surjektivität von F gibt es ein

$a \in M$ mit $F(a) = b$. Wir haben also $v_{\mathcal{N},(\beta_x^a)\circ F)}(A_0) = 1$. Nach Induktionsvoraussetzung folgt $v_{\mathcal{M},\beta_x^a}(A_0) = 1$ und damit auch $v_{\mathcal{M},\beta}(\exists x A_0) = 1$ folgt.

Gelte jetzt $v_{\mathcal{N},\beta\circ F}(A) = 0$. Wäre $v_{\mathcal{M},\beta}(A) \neq 0$, dann gäbe es ein Element $a \in M$ mit $v_{\mathcal{M},\beta_x^a}(A_0) \neq 0$. Aus der Induktionsvoraussetzung, $v_{\mathcal{N},(\beta_x^a\circ F}(A_0) \leq_i v_{\mathcal{M},\beta_x^a}(A_0)$, folgt dann auch $v_{\mathcal{N},(\beta_x^a\circ F}(A_0) v_{\mathcal{N},(\beta\circ F)_x^{F(a}}(A_0) \neq 0$. Das aber ist ein Widerspruch, denn aus $v_{\mathcal{N},\beta\circ F}(A) = 0$ folgt für alle $b \in N$ $v_{\mathcal{N},(beta\circ F)_x^b}(A_0) = 0$.

■

zu Aufgabe 2.1.11 Zuerst überzeugt man sich, daß für alle $W \subseteq \{0,1\}$, mit $W \neq \emptyset$ gilt $W = \{w \in \{0,1\} \mid T(W) \leq_i w\}$. Daraus folgt dann

$U(T(W_1), \ldots, T(W_n)) = \{(w_1, \ldots, w_n) \mid w_j \in W_j$ für alle $1 \leq j \leq n\}$.

Also gilt

$f^+(W_1, \ldots, W_n) = \{f(w_1, \ldots, w_n) \mid w_1, \ldots, w_n \in U(T(W_1), \ldots, T(W_n))\}$

Hieraus ist jetzt $T(f^+(W_1, \ldots, W_n)) = f^*(T(W_1), \ldots, T(W_n))$ unmittelbar einzusehen.

■

Zu 2.2 Funktionale Vollständigkeit

zu Aufgabe 2.2.2(3) Sei v eine beliebige aussagenlogische Bewertung.
Dann gilt stets

$$v((A \wedge B) \vee C) \leq v(A \vee C)$$

und

$$v((A \wedge B) \vee C) \leq v(B \vee C)$$

und damit auch

$$v((A \wedge B) \vee C) \leq v((A \vee C) \wedge (B \vee C))$$

Zum Beweis der umgekehrten Beziehung beachte man, daß $v(A \wedge B = v(A)$ oder $v(A \wedge B) = v(B)$ sein muß. Somit ist auch $v((A \wedge B) \vee C) = v(A \vee C)$ oder $v((A \wedge B) \vee C) = v(B \vee C)$, also in jedem Fall $v((A \wedge B) \vee C) \geq v((A \vee C) \wedge (B \vee C))$.

zu Aufgabe 2.2.2(9) Wir bemerken zunächst, daß für jede aussagenlogische Bewertung v gilt:

$$\text{wenn} \qquad v(A) \leq v(B), \qquad \text{dann} \qquad v(\neg B) \leq v(\neg A)$$

und

$$\text{wenn} \qquad v(A) < v(B), \qquad \text{dann} \qquad v(\neg B) < v(\neg A)$$

Wir unterscheiden die beiden Fälle:

1. $v(\neg A \wedge \neg B) = v(\neg A)$, d. h. $v(\neg A) \leq v(\neg B)$
2. $v(\neg A \wedge \neg B) = v(\neg B)$, d. h. $v(\neg B) \leq v(\neg A)$

Im Fall 1 folgt $v(B) \leq v(A)$ und damit $v(A \vee B) = v(A)$. Also auch $v(\neg(A \vee B)) = v(\neg A)$. Analog folgt im Fall 2 $v(\neg(A \vee B)) = v(\neg B)$.

zu Aufgabe 2.2.4 $J_k(x) = T_{m-1}(x + m - k - 1)$

zu Aufgabe 2.2.5 Die Notwendigkeit des Kriteriums ist klar. Zum Beweis der umgekehrten Implikation nehmen wir an, daß die Funktion g die Bedingung des Kriteriums erfüllt. Für jedes n-Tupel $\bar{a} =< a_1, \ldots, a_n >$ von Elementen aus M gibt es somit eine H-definierbare Funktion $h_{\bar{a}}(x_1, \ldots, x_n)$, so daß $g(\bar{a}) = h_{\bar{a}}(\bar{a})$. Dann gilt für

$$f_{\bar{a}}(x_1, \ldots, x_n) = \min(J_{a_1}(x_1), \ldots, J_{a_n}(x_n), h_{\bar{a}}(\bar{x}))$$

die Beziehung

$$\begin{aligned} f_{\bar{a}}(\bar{b}) &= 0 \text{ falls} \qquad \bar{a} \neq \bar{b} \\ f_{\bar{a}}(\bar{a}) &= g(\bar{a}) \end{aligned}$$

Ist $\bar{a}_1, \ldots, \bar{a}_k$ eine Aufzählung aller n-Tupel von M und schreiben wir vereinfachend f_i anstelle von $f_{\bar{a}_i}$, so gilt für die Funktion

$$f(\bar{x}) = \max(f_1(\bar{x}), \ldots, f_k(\bar{x}))$$

schließlich

$$f(\bar{b}) = g(\bar{b})$$

für alle n-Tupel $\bar{b}$.

zu Aufgabe 2.2.6 Wir schreiben suggestiv $x_1 + \ldots + x_k$ für die Hintereinadnerasuführung von g $g(x_1, \ldots g(x_{k-1}, x_k) \ldots)$ und entsprechend $x_1 \cdot \ldots \cdot x_k$ für die Hintereinanderausführung $f(x_1, \ldots f(x_{k-1}, x_k) \ldots)$
Man überzeugt sich als erstes, daß die Gültigkeit der Gleichung

$$\begin{aligned} h(x_1, \ldots, x_n) &= (J_0(x_1) \cdot f(0, x_2, \ldots, x_n) + \ldots \\ &\quad +(J_{m-1}(x_1) \cdot f(m-1, x_2, \ldots, x_n)) \end{aligned}$$

nur von den definierenden Eigenschaften der Funktionen f und g abhängt.
Danach zeigt man durch Induktion nach n, daß h in H definierbar ist. Für $n = 1$ folgt das unmittelbar aus der angegebenen Gleichung, weil in H die konstanten Funktionen zur Verfügung stehen. Der Induktionsschritt ist dann einfach.

zu Aufgabe 2.2.7 Die Lösung ist im wesentlichen eine abstrakte Version des Beweises von Lemma 4.

Seien $\{a_0, \ldots, a_{m-1}\}$ die Elemente der Algebra $\mathcal{M}$.

Wir beweisen die Aussage durch Induktion nach k. Nehmen also an, daß alle k-stelligen Funktionen schon definierbar seien und nehmen uns eine $k+1$-stellige Funktion f vor. Die Funktionen f_i, $0 \leq i \leq k+1$ seien definiert durch

$$f_i(x_1, \ldots, x_k) = f(x_1, \ldots, x_k, a_i)$$

Nach Induktionsvoraussetzung gibt es Terme t_i der Algebra, die diese Funktionen definieren. Nach Voraussetzung können wir davon ausgeghen, daß $k \geq 2$ ist. Also sind insbesonder alle 1-stelligen Funktionen definierbar, also auch die folgenden

$$J_i(x) = \begin{cases} a_{m-1} & \text{falls } x = a_i \\ a_0 & \text{sonst} \end{cases}$$

Seien t_i definierende Funktionsterme für J_i.

Darüber hinaus gibt es Terme t_{min} und t_{max} welche die folgenden beiden 2-stelligen Funktionen definieren:

$$\begin{array}{ll} max(x,y) = a_j & \text{falls } x = a_i, y = a_j \text{ und } i \leq j \\ min(x,y) = a_j & \text{falls } x = a_i, y = a_j \text{ und } j \leq i \end{array}$$

Wir schreiben $t_{max}(x_1, t_{max}(x_2, x_3))$ einfacher als $t_{max}(x_1, x_2, x_3)$.

$f(x_1, \ldots, x_k, x_{k+1})$ wird jetzt definiert durch

$$t_{max}(t_{min}(t_0, J_0(x_{k+1})), \ldots, t_{min}(t_{m-1}, J_{m-1}(x_{k+1})))$$

zu Aufgabe 2.2.8 Das in Lemma 4 beschriebene Verfahren liefert

$$\begin{array}{lll} A \Rightarrow B & id & A \wedge \sim\sim \neg B \qquad\qquad\qquad \vee \\ & & (\sim \neg A \vee u) \wedge \sim B \wedge \sim \neg B \quad \vee \\ & & (A \vee \sim A) \wedge \sim\sim B \end{array}$$

was sich vereinfachen läßt zu:

$$\begin{array}{lll} A \Rightarrow B & id & A \wedge \sim\sim \neg B \quad \vee \\ & & \sim B \wedge \sim \neg B \quad \vee \\ & & \sim\sim B \end{array}$$

zu Aufgabe 2.2.9 Die Funktionenmenge $\Sigma = \{0, \neg, \leftrightarrow\}$ erfüllt alle Voraussetzungen von Satz 9, reicht aber nicht aus um alle zwei-wertigen Funktionen darzustellen. Man zeigt, daß für jede aussagenlogische Formel, A, im Vokabular Σ gilt

1. A ist entweder logisch äquivalent zu 0 oder zu einer Formel B, in der 0 nicht vorkommt

2. es gibt eine zu A logisch äquivalente Formel der Form B oder $\neg B$, so daß in B weder 0 noch $\neg$ vorkommt.
3. jede Formel, die nur mit $\leftrightarrow$ und den beiden Aussagenvariablen X, Y aufgebaut ist, ist logisch äquivalent zu 1, X, Y oder $X \leftrightarrow Y$.

Zum Beweis der dritten Eigenschaft macht man am besten Gebrauch von der Beobachtung, daß eine Formel, die nur mit $\leftrightarrow$ aufgebaut ist, genau dann eine Tautologie ist, wenn jede Aussagenvariablen in ihr eine gerade Anzahl von Vorkommen hat.

Zu 2.3 Kalküle

zu Aufgabe 2.3.1

$$\frac{\{T,U\}A \wedge B}{\begin{array}{c}\{T,U\}A\\ \{T,U\}B\end{array}} \qquad \frac{\{T,U\}A \vee B}{\{T,U\}A \quad \text{oder} \quad \{T,U\}B}$$

$$\frac{\{T,U\} \sim A}{\{F,U\}A} \quad \frac{\{T,U\}\neg A}{\{U,F\}A} \quad \frac{\{T,U\}A \supset B}{\{T,U\}B \quad \text{oder} \quad \{F,U\}A}$$

$$\frac{\{T,U\}\forall x A(x)}{\{T,U\}A(z)} \quad \frac{\{T,U\}\exists x A(x)}{\{T,U\}A(f(y_1,\ldots,y_k))}$$

zu Aufgabe 2.3.2

$$\frac{\{T\}A \text{ cand } B}{\begin{array}{c}\{T\}A\\ \{T\}B\end{array}} \qquad \frac{\{F\}A \text{ cand } B}{\{F\}A \quad \text{oder} \quad \begin{array}{c}\{F\}B\\ \{T\}A\end{array}}$$

$$\frac{\{U\}A \text{ cand } B}{\{U\}A \quad \text{oder} \quad \begin{array}{c}\{T\}A\\ \{U\}B\end{array}} \qquad \frac{\{F,U\}A \text{ cand } B}{\{F,U\}A \quad \text{oder} \quad \{F,U\}B}$$

$$\frac{\{T\}A \text{ cor } B}{\{T\}A \quad \text{oder} \quad \begin{array}{c}\{T\}B\\ \{F\}A\end{array}} \qquad \frac{\{F\}A \text{ cor } B}{\begin{array}{c}\{F\}A\\ \{F\}B\end{array}}$$

$$\frac{\{U\}A \text{ cor } B}{\{U\}A \quad \text{oder} \quad \begin{array}{c}\{F\}A\\ \{U\}B\end{array}} \qquad \frac{\{F,U\}A \text{ cor } B}{\{U\}A \quad \text{oder} \quad \begin{array}{c}\{F\}A\\ \{U,F\}B\end{array}}$$

zu Aufgabe 2.3.3

$$\frac{\{T\}A \circ B}{\{U,F\}A \quad \textit{oder} \quad \begin{array}{c}\{F\}A\\ \{T\}B\end{array} \quad \textit{oder} \quad \begin{array}{c}\{F\}A\\ \{F\}B\end{array}} \qquad \frac{\{F\}A \circ B}{\begin{array}{c}\{F\}A\\ \{U\}B\end{array}}$$

Die Regel für $\{F,U\}A \circ B$ hat dieselbe Konklusion wie die Regel für $\{F\}A \circ B$. Für $\{U\}A \circ B$ gibt es keine Regel.

zu Aufgabe 2.3.4 Falls es eine Tableauregel für $\{1\}c(P_1, \ldots, P_n)$ mit den genannten Eigenschaften gibt, dann sieht man leicht, daß $c(P_1, \ldots, P_n)$ 1-persistent ist.
Nehmen wir umgekehrt an, daß $c(P_1, \ldots, P_n)$ 1-persistent ist, und betrachten ein n-Tupel $(w_1, \ldots, w_n)$ von Wahrheitswerten aus $\{1, u, 0\}$ mit $v(c(w_1, \ldots, w_n)) = 1$. Falls für ein i gilt $w_i = u$, dann gilt wegen der 1-Persistenz auch

$$\begin{array}{rcl} v(c(w_1, \ldots, w_{i-1}, 1, w_{i+1}, \ldots, w_n)) & = & 1 \\ v(c(w_1, \ldots, w_{i-1}, 0, w_{i+1}, \ldots, w_n)) & = & 1 \end{array}$$

. Seien $w^1 = (w_1^1, \ldots, w_n^1), \ldots, w^r = (w_1^r, \ldots, w_n^r)$ alle Belegungen der Aussagenvariablen $P_1, \ldots, P_n$ mit $v(c(w^j) = 1$. Wir setzen $E_j = \{\{w_i\}P_i \mid 1 \leq i \leq n \text{ und } w_i \neq u\}$. Man sieht leicht, daß

$$\frac{\{1\}c(P_1, \ldots, P_n)}{E_1 \mid \ldots \mid \quad E_r}$$

eine korrekte und vollständige Tableauregel ist, die als Vorzeichen nur $\{1\}$ und $\{0\}$ enthält.

Zu 2.5 Unendlichwertige Logiken

zu Aufgabe 2.5.1

$$\begin{array}{lcl@{\quad}c@{\quad}l} A \vee B & = & max\{A, B\} & = & (A \Rightarrow B) \Rightarrow B \\ A \wedge B & = & min\{A, B\} & = & \neg(\neg A \vee \neg B) \\ A \equiv B & = & 1- \mid A - B \mid & = & (A \Rightarrow B) \wedge (B \Rightarrow A) \end{array}$$

$$\begin{array}{lcl} A \vee B & = & \neg(\neg A \sqcup B) \sqcup B \\ A \wedge B & = & \neg(\neg A \vee \neg B) \\ A \Rightarrow B & = & \neg A \sqcup B \end{array}$$

Herleitungen:

$$\begin{array}{lcl} \neg(\neg A \sqcup B) \sqcup B & = & min\{1, 1 - min\{1, 1 - A + B\}, B\} \\ & = & min\{1, 1 + max\{-1, -1 + A - B\}, B\} \\ & = & min\{1, max\{B, \}\} \\ & = & max\{B, A\} \\ & = & A \vee B \end{array}$$

$$\begin{array}{lcl} \neg(\neg A \vee \neg B) & = & 1 - max\{1 - A, 1 - B\} \\ & = & 1 + min\{A - 1, B - 1\} \\ & = & min\{A, B\} \\ & = & A \wedge B \end{array}$$

$$\begin{array}{lcl} \neg A \sqcup B & = & min\{1, 1 - A + B\} \\ & = & A \Rightarrow B \end{array}$$

zu Aufgabe 2.5.2 Der Wahrheitswerteverlauf der Teilformel $\bigsqcap^k \neg A$ wird durch die Funktion $max\{0, (k-1) - kA\}$ beschrieben. Die Teilformel $A \sqcap \bigsqcup^{k-1}$ hat den Werteverlauf $max\{0, min\{A, kA - 1\}\}$. Die angegebene Formel ist eine Tautologie wenn für alle $A \in W_n$ gilt

$$min\{1, (n-1)(max\{0, (k-1) - kA\} + max\{0, min\{A, kA - 1\}\})\} = 1$$

i.e. wenn

$$max\{0, (k-1) - kA\} + max\{0, min\{A, kA - 1\}\} \geq \frac{1}{n-1}$$

Speziell für $k = 2$ führt das zu:

$$max\{0, 1 - 2A\} + max\{0, min\{A, 2A - 1\}\} \geq \frac{1}{n-1}$$

Für $(1-2A) > 0$ vereinfacht sich die linke Seite der Ungleichung zu $1-2A$. Wenn für $A \in W_n$ die Ungleichung $(1-2A) > 0$ gilt, dann gilt auch schon $(1-2A) \geq \frac{1}{n-1}$.
Falls $(1-2A) < 0$ so berechnet sich die linke Seite zu $min\{A, 2A-1\}$. Wegen $A \geq \frac{1}{n-1}$ und $2A-1 \geq \frac{1}{n-1}$ folgt auch in diesem Fall die Gültigkeit der Ungleichung.
Der noch verbleibende Fall $2A = 1$ ist für $A \in W_n$ ausgeschlossen, da 2 kein Teiler von $n-1$ sein darf.

zu Aufgabe 2.5.3

$$\begin{array}{rcl} p \Rightarrow^u q & = & max(0, p-q) \\ p \equiv^u q & = & |p-q| \end{array}$$

zu Aufgabe 2.5.4 Die gesuchte Funktion ist:

$$f^m(A) = \begin{cases} 1-A & \text{falls} \quad A \leq \frac{1}{m} \\ (m-1)A & \text{falls} \quad \frac{1}{m} \leq A \leq \frac{1}{m-1} \\ 1 & \text{falls} \quad \frac{1}{m-1} \leq A \end{cases}$$

zu Aufgabe 2.5.5 Die gesuchte Funktion ist:

$$g^m(A) = \begin{cases} A & \text{falls} \quad A \leq \frac{m-2}{m-1} \\ (2-m)+mA & \text{falls} \quad \frac{m-2}{m-1} \leq A \leq \frac{m-2}{m-1} \\ m-mA & \text{falls} \quad \frac{m-2}{m-1} \leq A \end{cases}$$

zu Aufgabe 2.5.6 Sei I eine unendliche Teilmenge von $\mathbb{N}$ und F eine Formel mit $F \in Taut(\mathcal{L}^n_{\text{Łukas}})$ für alle $n \in I$. Seien $X_1, \ldots, X_k$ die aussagenlogischen Atome in F. Dann gilt also für jede Belegung v mit $v(X_i) = q_i$, wobei alle q_i in $W_I = \bigcup_{n \in I} W_n$ liegen $v(F) = 1$. Sei jetzt $\bar{r} = (r_1, \ldots, r_k)$ eine beliebiges Tupel rationaler Zahlen. Da wegen der Unendlichkeit von I die Menge W_I dicht liegt in $\mathbb{Q}$ gibt es eine Folge $\bar{q}^m$ von k-Tupel in W_I^k die gegen $\bar{r}$ konvergiert. Wegen der Stetigkeit der durch F erzeugten Funktion gilt dann auch $w(F) = 1$ für die Belegung w mit $w(X_i) = r_i$.

zu Aufgabe 2.5.7

1. $$\begin{array}{lcll} 0 & \leq & f(A,0) & \text{nach Festlegung der Wahrheitswertemenge} \\ f(A,0) & = & f(0,A) & \text{Definition 29(1)} \\ f(0,A) & \leq & f(0,1) & \text{Definition 29(2)} \\ f(0,1) & \leq & 0 & \text{Definition 29(3)} \end{array}$$

2. $$\begin{array}{lcll} f(A,A) & \leq & f(A,1) & \text{Definition 29(2)} \\ f(A,1) & = & A & \text{Definition 29(3)} \end{array}$$

zu Aufgabe 2.5.8 Natürlich wird die Behauptung

$$F^k(A) = \frac{A}{2^k - (2^k - 1)A}$$

durch Induktion über k bewiesen.

Induktionsanfang $k = 1$

$F(A) = f(A, A) = \frac{A*A}{A+A-A*A} = \frac{A}{2-A}$

Induktionsschritt von k nach $k + 1$

$$\begin{array}{lcll} F^{k+1}(A) & = & F(F^k(A)) & \text{Definition von } F^{k+1} \\ & = & \frac{F^k(A)}{2-F^k(A)} & \text{Formel für } F(X) \\ & = & \frac{\frac{A}{2^k-(2^k-1)A}}{2-\frac{A}{2^k-(2^k-1)A}} & \text{Induktionsvor.} \\ & = & \frac{\frac{A}{2^k-(2^k-1)A}}{\frac{2(2^k-(2^k-1)A)-A}{2^k-(2^k-1)A}} & \text{Elementare Rechnung} \\ & = & \frac{A}{2^{k+1}-2^{k+1}A+2A-A} & \text{Elementare Rechnung} \\ & = & \frac{A}{2^{k+1}-(2^{k+1}-1)A} & \text{Elementare Rechnung} \end{array}$$

Zu 2.6Reduktion auf zweiwertige Logik

zu Aufgabe 2.6.1

zu Aufgabe 2.6.2 Sei $A^u =\sim A \wedge \sim \neg A$ dann gilt

$$v_{\mathcal{M}}(A) = u \Leftrightarrow v_{\mathcal{M}}(A^u) = 1.$$

Wähle man jetzt A_u als Übersetzung von A^u, also $A_u = (A^u)_0$, dann hat A_u offensichtlich die gewünschte Eigenschaft.

zu Aufgabe 2.6.3

Zu 2.7 Abstrakte Konsequenzrelationen

zu Aufgabe 2.7.1

zu Aufgabe 2.7.2 Gelte $\Sigma_1 \vdash A$ und $\Sigma_1 \subseteq \Sigma_2$. Für alle $B \in \Sigma_1$ gilt auch $B \in \Sigma_2$ und damit nach Axiom I auch $\Sigma_2 \vdash B$. Mit Axiom III folgt jetzt unmittelbar $\Sigma_2 \vdash A$.

▪

Zu 3.1 Modale Aussagenlogik

zu Aufgabe 3.1.1

1. $\Box(P \to Q) \to (\Box P \to \Box Q)$
   Sei $\mathcal{K} = (G, R, v)$ eine beliebige Kripkestruktur und $g \in G$ eine beliebig ausgewählte Welt. Wir nehmen $v(g, \Box(P \to Q) = 1$ an mit dem Ziel $v(g, \Box P \to \Box Q) = 1$ zu zeigen. Diesem Ziel kommen wir einen Schritt näher, wenn wir zusätzlich $v(g, \Box P) = 1$ annehmen. Jetzt bleibt noch $v(g, \Box Q) = 1$ zu zeigen. Dazu betrachten wir eine beliebige Welt $h \in G$ für die $R(g, h)$ gilt. Wenn wir für jedes solche h zeigen können, daß $v(h, Q) = 1$ gilt, dann sind wir fertig. Aus der zweiten Vorraussetzung $v(g, \Box P) = 1$ wissen wir auf jeden Fall $v(h, P) = 1$. Damit folgt aus der ersten Annahme, $v(g, \Box(P \to Q) = 1$, aber auch schon $v(h, Q) = 1$.
2. $\Box(P \land Q) \leftrightarrow (\Box P \land \Box Q)$
3. $\Diamond(P \lor Q) \leftrightarrow (\Diamond P \lor \Diamond Q)$
4. $(\Box P \lor \Box Q) \to \Box(P \lor Q)$
5. $\Diamond(P \land Q) \to (\Diamond P \land \Diamond Q)$

zu Aufgabe 3.1.2

1. Die Kripke Struktur $\mathcal{K}$ bestehe aus den drei Welten $G = \{g_1, g_2, g_3\}$. Jede Welt sei von jeder anderen aus zugänglich, i.e. R ist die universelle Relation auf G und v sei gegeben durch
   $v(g_i, P) = \begin{cases} 1 & \text{falls } i = 1 \text{ oder } i = 2 \\ 0 & \text{sonst} \end{cases}$
   $v(g_i, Q) = \begin{cases} 1 & \text{falls } i = 1 \text{ oder } i = 3 \\ 0 & \text{sonst} \end{cases}$
   Es gilt $(\mathcal{K}, g_1) \models \Box(P \vee Q)$. Aber es gilt weder $(\mathcal{K}, g_1) \models \Box P$ noch $(\mathcal{K}, g_1) \models \Box Q$.
2.

zu Aufgabe 3.1.3

zu Aufgabe 3.1.4

Aus $\vdash_G A \rightarrow B$ folgt noch $A \vdash_G B$

Ein Gegenbeispiel für die umgekehrte Implikationsrichtung erhält man für $A = p$ und $B = \Box p$. Denn es gilt $p \vdash_G \Box p$, aber nicht $\vdash_G p \rightarrow \Box p$.

Zu 3.2 Korrespondenztheorie

zu Aufgabe 3.2.1

1	reflexiv	$\Box p \rightarrow p$
2	symmetrisch	$p \rightarrow \Box\Diamond p$
3	seriell	$\Box p \rightarrow \Diamond p$
4	transitiv	$\Box p \rightarrow \Box\Box p$
5	Euklidisch	$\Diamond p \rightarrow \Box\Diamond p$
6	schwach funktional	$\Diamond p \rightarrow \Box p$
7	funktional	$\Diamond p \leftrightarrow \Box p$
8	dicht	$\Box\Box p \rightarrow \Box p$
9	schwach zusammenhängend	$\Box((p \wedge \Box p) \rightarrow q) \vee \Box((q \wedge \Box q) \rightarrow p)$
10	schwach gerichtet	$\Diamond\Box p \rightarrow \Box\Diamond p$
11	konfluent	$\Diamond\Box p \rightarrow \Diamond p$

zu 8 Zur Erinnerung wiederholen wir die definiertende Formel für Dichtheit: $\forall x \forall y(R(x,y) \rightarrow \exists z(R(x,z \wedge R(z,y))$
Sei zunächst $\mathcal{K} = (G,R,v)$ eine Kripke Struktur mit dichtem Rahmen (G,R). Wir wollen für alle $g \in G$ zeigen, daß $g \models \Box\Box p \rightarrow \Box p$ gilt. Gelte also

$$g \models \Box\Box p \tag{5.1}$$

Um $g \models \Box p$ zu zeigen, betrachten wir ein $h \in G$ mit $R(g,h)$. Aus der Dichtheit folgt die Existenz einer Welt $k \in G$ mit $R(g,k)$ und $R(k,h)$. Dann folgt aus 5.1 unmittelber $h \models \Box p$, wie gewünscht.

Sei jetzt, umgekehrt, (G,R) ein Kripke Rahmen, so daß für alle v und alle $g \in G$ gilt $((G,R,v),s) \models \Box\Box p \rightarrow \Box p$. Angenommen (G,R) ist nicht dicht. Dann gibt es g_0, g_1 in G, so daß kein $g \in G$ existiert, was dazwischen liegt, d.h. für das $R(g_0,g)$ und $R(g,g_1)$ gilt. Wir definieren v wie folgt

$$v(h,p) = \begin{cases} 1 & \text{falls ein } g \in G \text{ existiert mit } R(g_0,g) \text{ und } R(g,g_1). \\ 0 & \text{sonst} \end{cases}$$

Offensichtlich gilt $((G,R,v),g_0) \models \Box\Box p$ und $((G,R,v),g_0) \models \neg\Box p$. Also $((G,R,v),g_0) \not\models \Box\Box p \rightarrow \Box p$, im Widerspruch zur Voraussetzung.

zu 9 Schwach zusammenhängende Rahmen werden durch die Formel $\forall x \forall y \forall z(R(x,y) \wedge R(x,z) \rightarrow (R(y,z) \vee y = z \vee R(z,y))$ definiert.
Teil 1 Sei $\mathcal{K} = (G,R,v)$ eine Kripke Struktur mit schwach zusammenhängenden Rahmen (G,R). Angenommen es gäbe ein $s \in G$ mit

$$s \not\models \Box((p \wedge \Box p) \rightarrow q) \quad \text{und} \quad s \not\models \Box((q \wedge \Box q) \rightarrow p)$$

Dann gibt es s_1, t_1 mit $R(s,s_1)$ und $R(s,t_1)$ und

$$\begin{array}{ll} s_1 \models p & t_1 \models q \\ s_1 \models \Box p & t_1 \models \Box q \\ s_1 \models \neg q & t_1 \models \neg p \end{array}$$

Nach Voraussetzung muß $R(s_1,t_1)$ oder $R(t_1,s_1)$ oder $s_1 = t_1$ gelten. Der erste Fall folgt aus $s_1 \models \Box p$ jetzt $t_1 \models p$ im Widerspruch zu $t_1 \models \neg p$. Im zweiten Fall folgt aus $t_1 \models \Box q$ jetzt

$s_1 \models q$, ebenfalls im Widerspruch zu den bisher zusammengetragenen Annahmen. Der verbleibende Fall steht im Widerspruch zu $s_1 \models p$ und $t_1 \models \neg p$.

Teil 2 Sei (G, R) ein nicht schwach zusammenhängender Rahmen. Dann gibt es $s, s_1, t_1 \in G$ mit $R(s, s_1)$ und $R(s, t_1)$ aber

$$\neg R(s_1, t_1) \qquad \neg R(t_1, s_1) \qquad t_1 \neq s_1$$

Wir definieren eine Belegung v der aussagenlogischen Variablen durch

$$v(p, h) = \begin{cases} 1 & \text{falls } h = s_1 \text{ oder } R(s_1, h) \\ 0 & \text{sonst} \end{cases}$$

$$v(q, h) = \begin{cases} 1 & \text{falls } h = t_1 \text{ oder } R(t_1, h) \\ 0 & \text{sonst} \end{cases}$$

Aus dieser Festlegung folgt insbesondere $v(t_1, p) = 0$ und $v(s_1, q) = 0$. Man sieht leicht, daß $s_1 \models p \wedge \Box p$ gilt. Also insgesamt $s_1 \not\models (p \wedge \Box p) \to q$ und damit auch $s \not\models \Box(p \wedge \Box p) \to q$. Analog sieht man $s \not\models \Box(q \wedge \Box q) \to p$ ein. Insgesamt ist damit

$$(G, R, v) \not\models \Box((p \wedge \Box p) \to q) \vee \Box((q \wedge \Box q) \to p)$$

gezeigt.

zu Aufgabe 3.2.2
$(\Diamond p \wedge \Diamond q) \to (\Diamond(p \wedge \Diamond q) \vee \Diamond(q \wedge \Diamond p))$

zu Aufgabe 3.2.4
Teil 1 Nach Definition steht $C(0, 1, 2, 0)$ für

$$\forall w_1 \forall w_2 \forall w_3 ((w_1 = w_2 \wedge R^2(w_1, w_3)) \to \exists w_4 (R(w_2, w_4) \wedge w_3 = w_4))$$

Ersetzt man w_2 überall durch w_1 und w_4 überall durch w_3, so erhält man

$$\forall w_1 \forall w_3 (R^2(w_1, w_3) \to R(w_1, w_3)),$$

woraus durch Auflösung der Definition von $R^2(w_1, w_3)$ unmittelbar die Definition der Transitivität von R wird.

Teil 2 Nach Definition steht $C(0, 1, 0, 0)$ für

$$\forall w_1 \forall w_2 \forall w_3((w_1 = w_2 \wedge w_1 = w_3) \rightarrow \exists w_4(R(w_2, w_4) \wedge w_3 = w_4))$$

Ersetzt man w_2 und w_3 überall durch w_1, so erhält man nach trivialen Umformungen

$$\forall w_1 \exists w_4(R(w_1, w_4) \wedge w_1 = w_4)$$

also

$$\forall w_1 R(w_1, w_1)$$

Teil 3 Nach Definition steht $C(1, 1, 0, 0)$ für

$$\forall w_1 \forall w_2 \forall w_3((R(w_1, w_2) \wedge w_1 = w_3) \rightarrow \exists w_4(R(w_2, w_4) \wedge w_3 = w_4))$$

Ersetzt man w_3 und w_4 durch w_1 so erhält man nach trivialen Umformungen

$$\forall w_1 \forall w_2(R(w_1, w_2) \rightarrow R(w_2, w_1))$$

Teil 4 Nach Definition steht $C(1, 0, 1, 0)$ für

$$\forall w_1 \forall w_2 \forall w_3(R(w_1, w_2) \wedge R(w_1, w_3)) \rightarrow \exists w_4(w_2 = w_4 \wedge w_3 = w_4))$$

Hier genügt es w_4 durch w_3 zu ersetzen um wie gewünscht

$$\forall w_1 \forall w_2 \forall w_3(R(w_1, w_2) \wedge R(w_1, w_3) \rightarrow w_2 = w_3)$$

zu erhalten.

zu Aufgabe 3.2.5
Gilt für eine Welt g in einer Kripke Struktur $\mathcal{K} = (G, R, v)$ mit einem deterministischen Rahmen (G, R) $g \models \Box(p \vee q)$, dann gibt es entweder überhaupt keine Nachfolgewelt von g oder es gibt genau eine Nachfolgewelt g_1. Im ersten Fall gilt trivialerweise auch $\Box p$ und $\Box q$. Im zweiten Fall gilt $g_1 \models p \vee q$. Also gilt $g_1 \models p$ oder $g_1 \models q$. Und wegen der Eindeutigkeit von g_1 auch $g_1 \models \Box p$ oder $g_1 \models \Box q$.

Sei jetzt umgekehrt (G, R) ein Rahmen, so daß für alle v und alle $g \in G$ gilt $((G, R, v), g) \models \Box(p \vee q) \rightarrow (\Box p) \vee (\Box q)$. Angenommen (G, R) ist nicht deterministisch, dann gibt es g, g_1, g_2 mit $R(g, g_1)$, $R(g, g_2)$ und $g_1 \neq g_2$. Wir definieren eine Interpretation v durch $v(g_1, p) = 1$, $v(g_1, q) = 0$ $v(h, p) = 0$ und $v(h, q) = 1$ für alle $h \in G$, $h \neq g_1$. Offensichtlich gilt $g \models \Box(p \vee q)$, aber weder $g \models \Box p$ noch $g \models \Box q$.

zu Aufgabe 3.2.6
$\neg\Box p \rightarrow \Box\neg p$ charakterisiert, wie die Formel in Aufgabe 3.2.5, die Klasse aller deterministischen Rahmen.

Sei (G, R) also ein deterministischer Rahmen und gelte $g \models \neg\Box p$ für ein $g \in G$. Wir wollen $g \models \Box\neg p$ zeigen. Aus der Voraussetzung folgt, daß es ein h gibt mit $R(g, h)$ und $h \models \neg p$. Da h das einzige Element $x \in G$ ist mit $R(g, x)$, folgt auch $g \models \Box\neg p$, wie gewünscht.

Dasselbe Argument wie in der Lösung zu Aufgabe 3.2.5 zeigt auch die umgekehrte Implikation.

zu Aufgabe 3.2.7
Wir zeigen

$$(G_1, R_1, v_1, g) \models F \text{ gdw } (G_2, R_2, v_2, g) \models F.$$

durch strukturelle Induktion über den Aufbau der Formel F. Im Anfangsfall ist F eine aussagenlogische Variable p und die zu beweisende Aussage reduziert sich auf

$$v_1(g, p) = v_2(g, p)$$

was unmittelbar aus der Definition eines Unterrahmens folgt.

Von den Fällen im Induktionsschritt zeigen wir nur den Fall $\Box F$. Wir schreiben abkürzend $\mathcal{K}_i$ für (G_i, R_i, v_i).

Wir nehmen zu nächst an, daß $(\mathcal{K}_1, g) \models \Box F$ gilt. Das heißt, für alle $h \in G_1$ gilt $(\mathcal{K}_1, h) \models F$. Unser Ziel ist $(\mathcal{K}_2, g) \models \Box F$ nachzuweisen. Dazu berachten wir ein beliebiges $h \in G_2$ mit $R_2(g, h)$.

Wegen der Unterrahmeneigenschaft gilt dann auch $R_1(g, h)$ und wir erhalten aus dem eben gesagten $(\mathcal{K}_1, h) \models F$. Nach Induktionsvoraussetzung folgt daraus, wie gewünscht, $(\mathcal{K}_2, h) \models F$.

Nehmen wir jetzt umgekehrt an, daß $(\mathcal{K}_2, g) \models \Box F$ gilt. Daraus folgt für alle $h \in G_2$ $(\mathcal{K}_2, h) \models F$. Soweit die Voraussetzungen für diesen Fall. Wir zielen auf $(\mathcal{K}_1, g) \models \Box F$. Dazu müssen wir ein beliebiges Element $h \in G_1$ betrachten mit $R_1(g, h)$ und $(\mathcal{K}_1, h) \models F$ zeigen. Aus der Abgeschlossenheit von G_2 in G_1 folgt $h \in G_2$ und dann auch $R_2(g, h)$. Aus der Voraussetzung für den aktuellen Beweisfall folgt $(\mathcal{K}_2, h) \models F$, woraus mit Induktionsvoraussetzung schließlich $(\mathcal{K}_2, h) \models F$ folgt.

zu Aufgabe 3.2.8

Angenommen es gäbe eine modallogische Formel F, welche die Klasse aller Rahmen (G, R) mit $(G, R) \models \exists x \exists y R(x, y)$ charakterisiert. Sei $G_1 = \{g_1, g_2, g_3\}$ und $G_2 = \{g_1\}$. Für die Zugänglichkeitsrelationen setzen wir $R_1(x, y) \Leftrightarrow x = g_2$ und $y = g_3$ während R_2 die leere Relation ist. Für eine beliebige Interpretationsfunktion v_1 gilt nach Annahme

$$(G_1, R_1, v_1, g_i) \models F$$

für alle $i \in \{1, 2, 3\}$. Offensichtlich ist (G_2, R_2) ein abgeschlossener Unterrahmen von (G_1, R_1) und für die Einschränkung v_2 von v_1 gilt nach Aufgabe 3.2.7 auch $(G_2, R_2, v_2, g_1) \models F$. Weil g_1 das einzige Element in G_2 ist folgt auch $(G_2, R_2, v_2) \models F$ und nach Wahl von F müßte in (G_2, R_2) die Formeln $\exists x \exists y R(x, y)$ gelten, was aber nicht der Fall ist. Die Annahme der Charakterisierbarkeit muß somit zurückgenommen werden.

zu Aufgabe 3.2.9

Sei $\mathcal{K} = (G, R, v)$ die disjunkte Summe der Kripke Strukturen $\mathcal{K}_1 = (G_1, R_1, v_1)$ und $\mathcal{K}_2 = (G_2, R_2, v_2)$.

Wir zeigen für jede Formel $F \in Fml_{ALmod}$ durch Induktion nach der Komplexität von F:

$$\mathcal{K} \models F \text{ gdw } \mathcal{K}_1 \models F \text{ oder } \mathcal{K}_2 \models F$$

Wir zeigen, dazu für jede Welt $g \in G$

$$(\mathcal{K}, g) \models F \text{ gdw } (\mathcal{K}_1, g) \models F \text{ oder } (\mathcal{K}_2, g) \models F$$

Da G die disjunkte Summe von G_1 und G_2 ist, $G = G_1 \uplus G_2$, läßt sich diese Behauptung in zwei Teile aufspalten:

1. für jede Welt $g \in G_1$ gilt $\quad (\mathcal{K}, g) \models F$ gdw $(\mathcal{K}_1, g) \models F$
2. für jede Welt $g \in G_2$ gilt $\quad (\mathcal{K}, g) \models F$ gdw $(\mathcal{K}_2, g) \models F$

Im Induktionsanfang ist F eine aussagenlogische Variable und die Behauptung folgt unmittelbar aus der Definition der Bewertgungsfunktion v.

Wir betrachten den Fall $F = \Box F_0$.

Aus der Definition der Zugänglichkeitsrelation R von $\mathcal{K}$ folgt für $g \in G_i$

$$\text{für alle } h \in G: \quad R(g, h) \Leftrightarrow h \in G_i \wedge R_i(g, h)$$

Aus dieser Beobachtung und der Induktionsvoraussetzung folgt unmittelbar

1. für jede Welt $g \in G_1$ gilt $\quad (\mathcal{K}, g) \models \Box F_0$ gdw $(\mathcal{K}_1, g) \models \Box F_0$
2. für jede Welt $g \in G_2$ gilt $\quad (\mathcal{K}, g) \models \Box F_0$ gdw $(\mathcal{K}_2, g) \models \Box F_0$

■

zu Aufgabe 3.2.10
Sei $G_1 = \{g_1\}$, $G_2 = \{g_2\}$ mit $g_1 \neq g_2$. Außerdem gelte $R_i(g_i, g_i)$ für $i = 1, 2$. Angenommen es gäbe eine modale Formel F, welche Kripke Rahmen mit der Eigenschaft $\forall x \forall y R(x, y)$ charakterisiert. Dann gilt sicherlich für jede Kripke-Struktur $\mathcal{K}_i$ zum Rahmen G_i $\mathcal{K}_i \models F$. Nach Aufgabe 3.2.9 muß dann auch für die disjunkte Summe $\mathcal{K}$ von $\mathcal{K}_1$ und $\mathcal{K}_2$ gelten $\mathcal{K} \models F$. Das ist aber ein Widerspruch zu der Charakterisierungseigenschaft von F, da in G die Formel $\forall x \forall y R(x, y)$ nicht gilt.

■

zu Aufgabe 3.2.11
(1) Angenommen es gilt $(\mathcal{K}, g) \models \Box_a p$. Dann gilt für alle $g \in G$ mit $R_a(g, h)$ auch $(\mathcal{K}, h) \models p$. Da R_a als eine transitive Hülle insbesondere reflexiv ist, gilt auch $R_a(g, g)$ und damit schon einem $(\mathcal{K}, g) \models p$. Es bleibt also noch $(\mathcal{K}, g) \models \Box_a \Box_b p$ zu zeigen. Dazu betrachten wir $h_1, h_2 \in G$ mit $R_a(g, h_1)$ und $R_b(h_1, h_2)$. Da R_a die transitive Hülle von R_b ist gilt dann auch $R_a(g, h_2)$ und damit auch $(\mathcal{K}, h_2) \models p$.

zu Aufgabe 3.2.12 Sei (G, R) ein transitiver Rahmen, in dem die angegebene Formel $\Box(\Box p \to p) \to \Box p$ nicht gilt. Es gibt also eine Bewertungsfunktion v und $g_1 \in G$ mit $(\mathcal{K}, g_1) \not\models \Box(\Box p \to p) \to \Box p$ für $\mathcal{K} = (G, R, v)$. Wir wollen zeigen, daß daraus die Existenz einer unendlichen aufsteigenden Kette von Welten in G folgt. Aus der Annahme folgt zunächst

$$(\mathcal{K}, g_1) \models \Box(\Box p \to p) \tag{5.1}$$

und

$$(\mathcal{K}, g_1) \models \neg\Box p \tag{5.2}$$

Wegen 5.2 gibt es $g_2 \in G$ mit $R(g_1, g_2)$ und

$$(\mathcal{K}, g_2) \models \neg p \tag{5.3}$$

Wegen 5.1 gilt auch

$$(\mathcal{K}, g_2) \models \Box p \to p \tag{5.4}$$

Aus 5.4 und 5.3 zusammen folgt:

$$(\mathcal{K}, g_2) \models \neg\Box p \tag{5.5}$$

was die Existenz einer Welt $g_3 \in G$ zur Folge hat mit $R(g_2, g_3)$ und

$$(\mathcal{K}, g_3) \models \neg p \tag{5.6}$$

Wegen der vorausgesetzten Transitivität folgt auch $R(g_1, g_3)$ und damit nach 5.1 wieder

$$(\mathcal{K}, g_3) \models \Box p \to p \tag{5.7}$$

Jetzt ist leicht zu sehen, daß eine unendliche bzgl. R aufsteigenede Folge von Welten entsteht.
Sei jetzt umgekehrt eine Folge $g_i \in G$ gegeben mit $R(g_i, g_{i+1})$ für alle $i \geq 1$.Dazu definieren wir ein v durch:

$$v(h,p) = \begin{cases} 0 & \text{falls } R(h, g_i) \text{ für ein } i \text{ gilt.} \\ 1 & \text{sonst.} \end{cases}$$

Wir schreiben $\mathcal{K}$ für (G, R). Wegen $R(g_i, g_{i+1})$ gilt $(\mathcal{K}, g_i) \models \neg p$ und wegen $R(g_{i+1}, g_{i+2})$ auch $(\mathcal{K}, g_i) \models \neg\Box p$ für alle i.
Betrachten wir ein beliebiges $h \in G$. Gibt es kein i mit $R(h, g_i)$, dann gilt $v(h,p) = 1$ und damit auch $(\mathcal{K}, h) \models \Box p \to p$. Gilt dagegen $R(h, g_i)$ für ein i, so gilt wegen $(\mathcal{K}, g_i) \models \neg p$ jetzt $(\mathcal{K}, h) \models \neg\Box p$ und damit auch $(\mathcal{K}, h) \models \Box p \to p$. Insgesamt ist damit $(\mathcal{K}, h) \models \Box(\Box p \to p)$ für alle $h \in G$ gezeigt und $(\mathcal{K}, g_i) \models \neg\Box p$ für alle i. Die Formel $\Box(\Box p \to p) \to \Box p$ ist also nicht gültig in $\mathcal{K}$.

zu Aufgabe 3.2.13 Charakterisieren A_1 und A_2 beide die Klasse $\mathcal{R}$ von Kripke Rahmen dann gilt natürlich

$$\vdash_{\mathcal{R}} A_1 \leftrightarrow A_2$$

zu Aufgabe 3.2.15

zu Aufgabe 3.2.14

zu 1 $\Box p \to \Diamond p$

$$A^2 = \forall X \forall x (\forall y (R(x,y) \to X(y)) \to \exists z (R(x,z) \wedge X(z))))$$

$$\neg A^2 = \exists X \exists x (\forall y (R(x,y) \to X(y)) \wedge \forall z (\neg R(x,z) \vee \neg X(z)))$$

Skolemisiert

$$\forall y (\neg R(a,y) \vee p(y)) \quad \forall z (\neg R(a,z) \vee \neg p(z))$$

Resolution

$$\forall y(\neg R(a,y) \vee p(y)) \quad \forall z(\neg R(a,z) \vee \neg p(z)) \quad \forall u(\neg R(a,u))$$

Subsumption

$$\forall u(\neg R(a,u))$$

Anti-Skolemisierung

$$\exists x \forall u(\neg R(x,u))$$

Negation:

$$\forall x \exists u(R(x,u))$$

zu 2 $p \rightarrow \Box\Diamond p$

$$A^2 = \forall X \forall x(X(x) \rightarrow \forall y(R(x,y) \rightarrow \exists z(R(y,z) \wedge X(z))))$$

$$\neg A^2 = \exists X \exists x(X(x) \wedge \exists y(R(x,y) \wedge \forall z(\neg R(y,z) \vee \neg X(z))))$$

Skolemisiert

$$p(a) \quad R(a,b) \quad \forall z(\neg R(b,z) \vee \neg p(z))$$

Resolution

$$p(a) \quad R(a,b) \quad \forall z(\neg R(b,z) \vee \neg p(z)) \quad \neg R(b,a)$$

Ist se-äquivalent zu

$$R(a,b) \quad \neg R(b,a)$$

Anti-Skolemisierung

$$\exists x \exists y(R(x,y) \wedge \neg R(y,x))$$

Negation

$$\forall x \forall y(R(x,y) \rightarrow R(y,x))$$

zu 3 $\Diamond^m \Box^n p \rightarrow \Box^j \Diamond^k p$

$$
\begin{array}{l}
A^2 = \forall X \forall x(\\
\quad \exists y_1 \exists y_2 \ldots \exists y_m (R(x, y_1) \wedge R(y_1, y_2) \wedge R(y_{m-1}, y_m) \wedge \\
\quad \forall z_1 \forall z_2 \ldots \forall z_n (R(y_m, z_1) \rightarrow R(z_1, z_2) \rightarrow \ldots \rightarrow R(z_{n-1}, z_n) \rightarrow X(z_n))) \\
\rightarrow \\
\quad \forall u_1 \forall u_2 \ldots \forall u_j (R(x, u_1) \rightarrow R(u_1, u_2) \rightarrow R(u_{j-1}, u_j) \rightarrow \\
\quad \exists w_1 \exists w_2 \ldots \exists w_k (R(u_j, w_1) \wedge R(w_1, w_2) \wedge \ldots \wedge R(w_{k-1}, w_k) \wedge X(w_k))) \\
\quad)
\end{array}
$$

$$
\begin{array}{l}
\neg A^2 = \exists X \exists x(\\
\quad \exists y_1 \exists y_2 \ldots \exists y_m (R(x, y_1) \wedge R(y_1, y_2) \wedge R(y_{m-1}, y_m) \wedge \\
\quad \forall z_1 \forall z_2 \ldots \forall z_n (R(y_m, z_1) \rightarrow R(z_1, z_2) \rightarrow \ldots \rightarrow R(z_{n-1}, z_n) \rightarrow X(z_n))) \\
\wedge \\
\quad \exists u_1 \exists u_2 \ldots \exists u_j (R(x, u_1) \wedge R(u_1, u_2) \wedge R(u_{j-1}, u_j) \wedge \\
\quad \forall w_1 \forall w_2 \ldots \forall w_k (\neg R(u_j, w_1) \vee \neg R(w_1, w_2) \vee \ldots \vee \neg R(w_{k-1}, w_k) \vee \neg X(w_k))) \\
\quad)
\end{array}
$$

Die Implikationen werden nach rechts geklammert gelesen, d.h.
$A \rightarrow B \rightarrow C$ steht für $A \rightarrow (B \rightarrow C)$

Skolemisierung liefert:

$$
\begin{array}{l}
R(a, b_1) \quad R(b_1, b_2) \quad R(b_{m-1}, b_m) \\
\forall z_1 \forall z_2 \ldots \forall z_n (\neg R(b_m, z_1) \vee \ldots \vee \neg R(z_{n-1}, z_n) \vee p(z_n)) \\
\\
R(a, c_1) \quad R(c_1, c_2) \quad R(c_{j-1}, c_j) \\
\forall w_1 \forall w_2 \ldots \forall w_k (\neg R(c_j, w_1) \vee \ldots \vee \neg R(w_{k-1}, w_k) \vee \neg p(w_k))
\end{array}
$$

Resolution

$$
\begin{array}{l}
R(a, b_1) \quad R(b_1, b_2) \quad R(b_{m-1}, b_m) \\
\forall z_1 \forall z_2 \ldots \forall z_n (\neg R(b_m, z_1) \vee \ldots \vee \neg R(z_{n-1}, z_n) \vee p(z_n)) \\
R(a, c_1) \quad R(c_1, c_2) \quad R(c_{j-1}, c_j) \\
\forall w_1 \forall w_2 \ldots \forall w_k (\neg R(c_j, w_1) \vee \ldots \vee \neg R(w_{k-1}, w_k) \vee \neg p(w_k)) \\
\forall z_1 \forall z_2 \ldots \forall z_n \forall w_1 \forall w_2 \ldots \forall w_{k-1} \\
(\neg R(b_m, z_1) \vee \ldots \vee \neg R(z_{n-1}, z_n \vee \neg R(c_j, w_1) \vee \ldots \vee \neg R(w_{k-1}, z_n))
\end{array}
$$

Es ist wichtig zu bemerken, daß w_k mit z_n unifiziert wurde. Ist se-äquivalent zu

$$\begin{array}{l} R(a,b_1) \quad R(b_1,b_2) \quad R(b_{m-1},b_m) \\ R(a,c_1) \quad R(c_1,c_2) \quad R(c_{j-1},c_j) \\ \forall z_1 \forall z_2 \ldots \forall z_n \forall w_1 \forall w_2 \ldots \forall w_{k-1} \\ (\neg R(b_m,z_1) \vee \ldots \vee \neg R(z_{n-1},z_n) \vee \neg R(c_j,w_1) \vee \ldots \vee \neg R(w_{k-1},z_n)) \end{array}$$

Anti-Skolemisierung

$$\begin{array}{l} \exists x \exists y_1 \ldots \exists y_m \exists u_1 \ldots \exists u_j (\\ R(x,y_1) \wedge R(y_1,y_2) \wedge R(y_{m-1},y_m) \wedge \\ R(x,u_1) \wedge R(u_1,c_2) \wedge R(u_{j-1},u_j) \wedge \\ \forall z_1 \forall z_2 \ldots \forall z_n \forall w_1 \forall w_2 \ldots \forall w_{k-1} \\ (\neg R(y_m,z_1) \vee \ldots \vee \neg R(z_{n-1},z_n) \vee \neg R(u_j,w_1) \vee \ldots \vee \neg R(w_{k-1},z_n))) \end{array}$$

Negation

$$\begin{array}{l} \forall x \forall y_1 \ldots \forall y_m \forall u_1 \ldots \forall u_j (\\ R(x,y_1) \wedge R(y_1,y_2) \wedge R(y_{m-1},y_m) \wedge \\ R(x,u_1) \wedge R(u_1,c_2) \wedge R(u_{j-1},u_j) \to \\ \exists z_1 \exists z_2 \ldots \exists z_n \exists w_1 \forall w_2 \ldots \exists w_{k-1} \\ (R(y_m,z_1) \wedge \ldots \wedge R(z_{n-1},z_n) \wedge R(u_j,w_1) \wedge \ldots \wedge R(w_{k-1},z_n))) \end{array}$$

Mit der Notation aus Definition 44 liest sich das als

$$\forall x \forall y_m \forall u_j ((R^m(x,y_m) \wedge R^j(x,u_j)) \to \exists z_n (R^n(y_m,z_n) \wedge R^k(u_j,z_n)))$$

zu Aufgabe 3.2.16

zu 1 Sei $\mathcal{M}$ eine Struktur mit $\mathcal{M} \models r(f(c)) \wedge q(a) \wedge \neg q(b)$ und $f(a) = f(b)$. Wir müssen zeigen, daß es nicht möglich ist $\mathcal{M}$ durch die Interpretation von p so zu einer Struktur $\mathcal{M}^*$ zu erweitern, daß gilt $\mathcal{M}^* \models \forall x(q(x) \vee p(f(x)))$ und $\mathcal{M}^* \models \neg p(f(a))$. Da auch $\mathcal{M}^* \models \neg q(b)$ gilt muß auf jeden Fall $\mathcal{M}^* \models p(f(b))$. Wegen der Gleichheit $f(a) = f(b)$ also auch $\mathcal{M}^* \models p(f(a))$. Das ist aber ein Widerspruch.

zu 2 Die se-äquivalente Formel ohne p lautet:

$$r(f(c)) \wedge q(a) \wedge \neg q(b) \wedge \forall x(q(x) \vee f(x) \neq f(a))$$

zu Aufgabe 3.2.17 $p \leftrightarrow \Box p$ ist eine charakterisierende Formel.

Sei (G, R) ein Rahmen mit $\forall x, y(R(x, y) \leftrightarrow x = y$ und v beliebig, dann siehmt man leicht, daß für $\mathcal{K} = (G, R, v)$ gilt $\mathcal{K} \models p \leftrightarrow \Box p$.

Sei jetzt umgekehrt (G, R) ein Rahmen, so daß für alle v gilt $\mathcal{K} = (G, R, v) \models p \leftrightarrow \Box p$. Betrachten wir zunächst v_1 mit $v_1(g, p) = 0$ für alle $g \in G$. Dann gilt $(\mathcal{K}, g) \models \neg p$ für alle $g \in G$. Also muß für alle g mindestens ein h existieren mit $R(g, h)$, da sonst $(\mathcal{K}, g) \models \Box p$ gelten würde. Jetzt nehmen wir an, daß für ein g_0 eine Welt $g_1 \in G$ existiert mit $R(g_0, g_1)$ und $g_0 \neq g_1$. Wir definieren

$$v(x, p) = \begin{cases} 1 & \text{falls } x = g_0 \\ 0 & \text{sonst} \end{cases}$$

Offensichtlich gilt $(\mathcal{K}, g_0) \models p$, aber $(\mathcal{K}, g_0) \models \neg\Box p$. Dieser Widerspruch zeigt, daß die obige Annahme falsch wahr. Zusammenfassen können wir feststellen: Zu jedem $g \in G$ gibt es ein h mit $R(g, h)$, aber es gibt keine k mit $R(g, k)$ und $g \neq k$, d.h. R ist die Gleichheit auf G.

zu Aufgabe 3.2.18 Die Vermutung ist falsch. Hier ist ein Gegenbeispiel. Die Formeln $\Box p \to p$ und $\Box q \to q$ charakterisieren beide die Klasse der reflexiven Rahmen. Die Formeln $(\Box p \to p) \leftrightarrow (\Box q \to q)$ ist aber keine Tautologie.

zu Aufgabe 3.2.19 Wenn der Rahmen (G, R) reflexiv ist, dann gilt für jedes v $(G, R, v) \models \Box p \to p$. Nach dem Lemma 39 gilt dann $(G*, R*, v*) \models \Box p \to p$. Um einen Widerspruch zur Tatsache, daß $(G*, R*)$ nicht reflexiv ist zu erhalten, müßte man für jede Bewertungsfunktion w zeigen, daß $(G*, R*, w) \models \Box p \to p$ gilt. Da aber nicht jedes w von der Form $v*$ ist (insbesonder für w mit $w(g^1, p) \neq w(g^2, p)$) ergibt sich kein Widerspruch.

Zu 3.3 Vollständigkeit und kanonische Modelle

zu Aufgabe 3.3.1

zu Aufgabe 3.3.2

zu Aufgabe 3.3.3

zu Aufgabe 3.3.4

zu Aufgabe 3.3.5

Zu 3.4 Entscheidbarkeit modaler Aussagenlogiken

zu Aufgabe 3.4.1 Es genügt den Induktionsschritt von B auf $\Box B$ im Beweis von Lemma 58 mit R^Γ anstelle von R_Γ zu betrachten. Wir fixieren also B und $\Box B \in \Gamma$.

Wir bemerken zunächst, daß aus der Definition von R^Γ unmittelbar für alle $g, h \in G$ folgt: wenn $R(g,h)$, dann auch $R^\Gamma([g],[h])$. Das genügt schon um die Implikation von zu links nach rechts beweisen. Gilt nämlich $(\mathcal{K}_\Gamma, [g]) \models \Box B$. Dann gilt für alle $h \in G$ mit $[g]R^\Gamma[h]$ auch $(\mathcal{K}_\Gamma, [h]) \models B$. Betrachten wir jetzt $g, h \in G$ mit $R(g,h)$, dann folgt nach der obigen Beobachtung $R^\Gamma([g],[h])$ und damit $(\mathcal{K}_\Gamma, [h]) \models B$. Woraus mit der Induktionsvoraussetzung $(\mathcal{K}, h) \models B$ folgt. Dieses Argument gilt für alle h der betrachteten Art, also haben wir $(\mathcal{K}, g) \models \Box B$ nachgewiesen.

Gelte jetzt umgekehrt $(\mathcal{K}, g) \models \Box B$ und betrachten wir ein $h \in G$ mit $R^\Gamma([g],[h])$. Dann folgt unmittelbar aus der Definition von R^Γ schon $(\mathcal{K}, h) \models B$. Die Induktionsvoraussetzung ergibt, wie gewünscht, $(\mathcal{K}, [h]) \models B$. Da dieses Argument für jedes h der betrachteten Art gilt haben wir ingesamt $(\mathcal{K}_\Gamma, [g]) \models \Box B$ gezeigt.

■

zu Aufgabe 3.4.2 Angenommen $A \vdash_G^K B$ gilt nicht und $\mathcal{K}$ ist eine Kripke Struktur, so daß $(\mathcal{K}, g) \models A$ für alle g und $(\mathcal{K}, g_0) \models \neg B$ für eine Welt g_0 von $\mathcal{K}$. Ist Γ die Menge aller Teilformeln von B und A und $\mathcal{K}_\Gamma$ die Filtration von $\mathcal{K}$ modulo Γ, dann gilt nach

Lemma 58 für alle Welten $h = [g]$ von $\mathcal{K}_\Gamma$ wieder $(\mathcal{K}_\Gamma, h) \models A$ und $(\mathcal{K}_\Gamma, [g_0]) \models \neg B$. Also ist auch die endliche Kripke Struktur $\mathcal{K}_\Gamma$ ein Gegenbeispiel für $A \vdash_G^K B$. Aus diesen Überlegungen folgt wie in Satz 59 die Entscheidbarkeit der globalen Folgerungsbeziehung.

(a) Struktur $\mathcal{K}$ (b) Filtration $\mathcal{K}_\Gamma$

Abbildung 5.1: Filtration transitiver Strukturen

zu Aufgabe 3.4.3 Die Kripke Struktur $\mathcal{K}$ auf der linken Seite der Abbildung 5.1 ist transitiv, trivialerweise, da die Voraussetzungen der Transitivitätsforderung nicht gegeben sind. Auf der rechten Seite der Abbildung ist die Filtration von $\mathcal{K}$ bezüglich der Formelmenge $\Gamma = \{p, q\}$ zu sehen, mit $S_1 = \{s_1\}$,$S_2 = \{s_2, s_3\}$, $S_3 = \{s_4\}$. $\mathcal{K}_\Gamma$ ist nich transitiv, da die der Übergang von S_1 nach S_2 fehlt.

zu Aufgabe 3.4.4 Wegen Aufgabe 3.2.1 genügt es zu zeigen, daß jede Filtration einer Kripke Struktur mit dichter Zugänglichkeitsrelations wieder eine dichte Zugänglichkeitsrelations besitzt.

Zu 3.6 Modale Quantorenlogik

zu Aufgabe 3.6.1

zu Aufgabe 3.6.2 Sei $A = \forall x(q(x) \to \Box q(x)) \land \exists x(\neg q(x) \land \Diamond q(x))$. Sei $\mathcal{K} = (G, R, M)$ ein Modell von A, d.h. für jedes $g \in G$ gilt $(\mathcal{K}, g) \models A$. Sei g_0 ein beliebiges Element in G. Mit U wollen

wir das gemeinsame Universum aller prädikatenlogischen Strukturen $M(g)$ für $g \in G$ bezeichnen. Wegen $(\mathcal{K}, g_0) \models \exists x(\neg q(x) \wedge \Diamond q(x))$ gibt es $a_0 \in U$ und eine Welt g_1 mit $g_0 \models \neg q(a_0$ und $g_1 \models q(a_0$. Da für g_1 auch wieder $g_1 \models \exists x(\neg q(x) \wedge \Diamond q(x))$ gilt finden wir $a_1 \in U$ und $g_2 \in G$ mit $g_1 \models \neg q(a_1$ und $g_2 \models q(a_1$. So fortfahrend finden wir für jedes $i \in \mathbb{N}$ Elemente $a_i \in U$ und Welten g_i
inG mit $g_i \models \neg q(a_i$ und $g_{i+1} \models q(a_i$. Da nach Annahme außerdem für jedes i gilt $g_i \models \forall x(q(x) \rightarrow \Box q(x))$ folgt für alle i, j mit $i < j$ sowohl $g_i \models \neg q(a_j$ als auch $g_j \models q(a_i)$. Daraus folgt, daß für $i \neq j$ auch $g_i \neq g_j$ gelten muß.

zu Aufgabe 3.6.3 Nein, es gibt keine solche Formel; die im Beweis des Vollständigkeitssatzes konstruierte Struktur erfüllt die angegebene Einschränkung.

zu Aufgabe 3.6.4

zu Aufgabe 3.6.5

zu Aufgabe 3.6.6

1. $\forall x \forall z(r(y, x) \wedge r(x, z) \rightarrow p(z)) \rightarrow \forall z(r(y, z) \rightarrow p(z))$
2. $\forall x(r(y, x) \rightarrow \exists z(r(x, z) \wedge p(z))) \rightarrow \exists x(r(y, x) \wedge \forall z(r(x, z) \rightarrow p(z))$
3. $p(y) \rightarrow \forall x(r(y, x) \rightarrow p(x))$

zu Aufgabe 3.6.7 Sei c ein neues Konstantensymbol. Für Σ sei $Red_l(\Sigma) = \{B(c/y) \mid B \in \Sigma\}$. Dann behaupten wir

$$\begin{gathered} Red_l(\Sigma) \cup \Sigma_{Kripke} \cup \{w(c)\} \vdash_{PK} Red(A)(c/y) \\ \text{gdw} \\ \Sigma \vdash_L^{\mathbf{K}} A \end{gathered}$$

Der Beweis ist jetzt einfach einzusehen.

zu Aufgabe 3.6.8

(a) Im aussagenlogischen Fall können wir es einrichten, daß für alle A die übersetzte Formel $Red(A)$ nur die beiden Variablen x und y enthält und zwar genau eine davon als freie Variable, während die jeweils andere gebunden vorkommt. Die einzige Wahlfreiheit haben wir in den Klauseln 5 und 6 von Definition 72 bei der Wahl der neuen gebundenen Variablen. Ist A eine atomare Formel, d.h. $A \equiv p$ für eine aussagenlogische Variable p, dann ist $Red(A) \equiv p(x)$ und die Behauptung ist sicher erfüllt. Aussagenlogische Kombinationen zerstören die genannte Eigenschaft nicht. Sei jetzt A eine Formel, so daß $Red(A)$ die Variable x frei enthält und y gebunden. Dann wählen wir $Red(\Box A) \equiv \forall x(R(y, x) \rightarrow Red(A))$. Damit kommen in $Red(\Box A)$ tatsächlich nur die Variablen x und y vor, und zwar y also freie Variable und x als gebundene.

Analog verfährt man mit $\Diamond A$.

(b) Angenommen es gäbe eine Formel $\phi_{trans} \in FO^2$, so daß für alle Strukturen $\mathcal{M} = (M, R)$ gilt

$$\mathcal{M} \models \phi_{trans} \Leftrightarrow R \text{ ist transitiv}$$

Die Konjunktion Θ der Formeln

$$\begin{array}{l} \forall x \exists y (R(x, y)) \\ \forall x \neg R(x, x) \\ \phi_{trans} \end{array}$$

ist sicher erfüllbar (wähle M = Menge der natürlichen Zahlen und R die strikte Ordnung). Aber es gibt kein endliches Modell von Θ. An genommen $\mathcal{M}_0 = (M_0, r)$ wäre ein endliches Modell von Θ. Wähle $a_0 \in M_0$ beliebig. Wegen $\mathcal{M}_0 \models \forall x \exists y (R(x, y)$ gibt es $a_1 \in M_0$ mit $r(a_0, a_1)$. In dieser Weise fortfahrend finden wir $a_i \in M_0$ mit $r(a_{i-1}, a_i)$. Wegen der Endlichkeit von $\mathcal{M}_0$ können nicht alle a_i verschieden sein, d.h. es gibt i, j, $i < j$ mit $a_i = a_j = a$. Wegen der Transitivität von r gilt auch $r(a_i, a_j)$, d.h. $r(a, a)$. Das steht im Widerspruch zu $\mathcal{M}_0 \models \forall x \neg R(x, x)$.

Zu 3.7 Kalküle

zu Aufgabe 3.7.1 In der Behandlung des Falles einer Anwendung der π-Regel muß die Interpretationsfunktion I fortgesetzt werden, so daß σ' zum Definitionsbereich gehört. Dabei muß die Bedingung aus Definition 75

$$
\begin{array}{ll}
I(\mu)RI(\mu') & \text{gilt, falls } \mu' \text{ von } \sigma \text{ aus zugänglich ist,} \\
\text{und} & \\
I(\mu) \models A & \text{gilt für jede Formel } \sigma A \text{ auf dem Pfad } P.
\end{array}
$$

erhalten bleiben. Das gelingt nur wenn wir wissen, daß σ' nicht Anfangsstück eines schon vorhanden Präfix ist.

zu Aufgabe 3.7.1 Man überlegt sich leicht, daß für jedes Präfix σ, das auf einem Pfad P eines K- Tableaus vorkommt, auch jedes Anfangsstück von σ auf dem Pfad vorkommen muß.

Die Formel $\Box p \vee \neg\Diamond\Box p$ ist sicherlich keine Tautologie für die modale Logik S4.

$$
\begin{array}{rll}
1 & F\Box p \vee \neg\Diamond\Box p & (1) \\
1 & F\Box p & (2) \text{ aus } 1 \\
1 & F\neg\Diamond\Box p & (3) \text{ aus } 1 \\
1,2,3 & Fp & (4) \text{ aus } 2 \\
1 & T\Diamond\Box p & (5) \text{ aus } 3 \\
1,2 & T\Box p & (6) \text{ aus } 5 \\
1,2,3 & Tp & (7) \text{ aus } 6
\end{array}
$$

Dieses Tableaux ist geschlossen, was im Widerspruch zur Korrektheit des Kalküls stehen würden. Das vorliegende Tableau ist allerdings nicht korrekt gebildet. Um Zeile (6) zu erhalten wird die π-Regel angewendet, das dabei angeführte Präfix $1,2$ ist auch neu, aber es ist verbotenerweise Anfangsstück des schon existierenden Präfix $1,2,3$. Damit ist gezeigt, daß die Einschränkung an die Anwendung der π-Regel für die Korrektheit des Kalküls unabdingbar ist. Man könnte allerdings durch eine andere Vereinbarung denselben Zweck erreichen: Man verlangt, daß bei der Einführung eines neuen Präfix jeweils ein neues Präfix minimaler Länge eingeführt wird.

zu Aufgabe 3.7.3

1. Man sieht unmittelbar aus den Tableauregeln in Abschnitt 3.7.1, daß durch eine Regelanwendung nur Teilformeln von Formeln, die schon auf einem Ast vorkommen, eingeführt werden. (Das gilt so natürlich nur für die Aussagenlogik. Im prädikatenlogischen Fall entstehen durch die bei Pfadabschlüssen auftretenden Substitutionen Formeln, die zwar Instanzen von Teilformeln der Eingabeformel sind, nicht aber selbst Teilformeln sind.)
2. Für **K**-Tableaux ist σ_2 von σ_1 erreichbar, genau dann wenn $\sigma_2 = \sigma_{1k}$ für ein geeignetes k. Wir führen Induktion nach der Anzahl der Formeln in $F_{\sigma_2}(P)$. Im Induktionsanfang betrachten wir den Fall, daß der Präfix σ_2 durch die Anwendung einer π-Regel neu eingeführt wird. Die Regel wird angewandt auf eine Formel $\sigma_1 T\Diamond B \in P_0$ und liefert als Ergebnis eine neue Formel $\sigma_2 B \in P$. Sicherlich gilt

   $$\text{laenge}(B) < \text{laenge}(\Diamond B) \leq \max(\text{laenge}(A) \mid A \in F_{\sigma_1}(P))$$

   Im Induktionsschritt wird der Pfad P_0 verlängert zu P, indem eine Formel $\sigma_2 TC$ hinzugefügt wird. Die einzige Möglichkeit dazu ist eine Anwendung der ν-Regel auf eine Formel der Form $\sigma_1 T\Box C$. Es bleibt überhaupt nur etwas zu zeigen, wenn $\max(\text{laenge}(A) \mid A \in F_{\sigma_2}(P)) = \text{laenge}(C)$ ist. Dann folgt aber wieder

   $$\text{laenge}(C) < \text{laenge}(\Box C) \leq \max(\text{laenge}(A) \mid A \in F_{\sigma_1}(P))$$

   Man beachte, daß dieses Argument z.B. für eine transitive Erreichbarkeitsrelation nicht funktioniert.

zu Aufgabe 3.7.4

1. Der allgemeine Rahmen eines Korrektheitsnachweises ist durch Satz 67 und Lemma 68 vorgegeben. Betrachten wir den Kern des Arguments im Falle der **S4**-ν-Regel: gilt in einer *transitiven* Kripkestruktur in der Welt $I(\sigma)$ die ν-Formel $\Box\nu_0$ und ist $I(\sigma')$ eine Nachfolgerwelt, i.e. gilt $I(\sigma)RI(\sigma')$,

dann gilt $I(\sigma') \models \nu_0$ wie immer, aber wegen der Transitivität gilt zusätzlich $I(\sigma') \models \Box\nu_0$ und das ist gerade die Konklusion der **S4**-ν-Regel.

2. Nach Aufgabe 3.7.3 Teil 1 gibt es nur endliche viele Möglichkeiten für die Mengen $F_\sigma(P)$, so daß jeder Pfad nach endlich vielen Schritten entweder geschlossen sein muß oder die Abbruchbedingung erfüllt.

3. Sei P ein offener Pfad in dem **S4**-Tableau, der die Abbruchbedingung erfüllt. Wir konstruieren eine Kripke Struktur $\mathcal{K} = (G, R, v)$. G_0 sei zunächst die Menge aller auf P vorkommenden Präfixe. Wir betrachten die Äquivalenzrelation $\sim$ auf G_0, die durch folgende Relationen erzeugt wird

   - für alle Blätter σ_1, σ_2 des Präfixbaums mit $F_{\sigma_1}(P) = F_{\sigma_2}(P)$ gilt $\sigma_1 \sim \sigma_2$
   - ist σ' ein Blatt des Präfixbaums, σ'_2 ein Wiederholungspräfix und gilt für $\sigma'_2 < \sigma'_1 < \sigma'$ $F_{\sigma'}(P) = F_{\sigma'_1}(P)$, dann gilt $\sigma' \sim \sigma'_1$

   Die Äquivalenzklasse eines Elements $\sigma \in G_0$ bezügl. $\sim$ bezeichnen wir mit $[\sigma]$. Die Menge der Welten ist jetzt $G = \{[\sigma] \mid \sigma \in G_0\}$ Wir setzen für jede Aussagenvariable p und $\sigma \in G$

   $$v(p, [\sigma]) = \begin{cases} 1 & \text{falls } \sigma T p \in P \\ 0 & \text{falls } \sigma F p \in P \\ \text{beliebig} & \text{sonst} \end{cases}$$

   Man beachte, daß diese Definition unabhängig von der Wahl des Repräsentanten der Äquivalenzklasse ist.

   Sei zunächst R_0 die echte Teilfolgenrelation auf G_0. R_1 sei der transitive Abschluß von $R_0 \cup \sim$ und $R([\sigma_1], [\sigma_2])$ gelte genau dann, wenn $R_1(\sigma_1, \sigma_2)$.

   Wir zeigen jetzt für die so festgelegte Kripkestruktur $\mathcal{K}$

   $$\text{wenn } \sigma\phi \in P \text{ dann } (\mathcal{K}, [\sigma]) \models \phi$$

   durch Induktion über die Komplexität der Formel ϕ.

zu Aufgabe 3.7.5

1	$F\Box\Diamond p \wedge \Diamond p \rightarrow \Diamond\Box p$	(1)
1	$T\Box\Diamond p \wedge \Diamond p$	(2)[1]
1	$F\Diamond\Box p$	(3)[1]
1	$T\Box\Diamond p$	(4)[2]
1	$T\Diamond p$	(5)[2]
11	Tp	(6)[5]
11	$T\Diamond p$	(7)[4]
11	$T\Box\Diamond p$	(8)[4]
11	$F\Box p$	(9)[3]
11	$F\Diamond\Box p$	(10)[3]
111	Tp	(11)[7]
111	$T\Diamond p$	(12)[8]
111	$T\Box\Diamond p$	(13)[8]
111	$F\Box p$	(14)[10]
111	$F\Diamond\Box p$	(15)[10]

112	Fp	(16)[9]
112	$T\Diamond p$	(17)[8]
112	$T\Box\Diamond p$	(18)[8]
112	$F\Box p$	(19)[10]
112	$F\Diamond\Box p$	(20)[10]

1121	Tp	(21)[17]
1121	$T\Diamond p$	(22)[18]
1121	$T\Box\Diamond p$	(23)[18]
1121	$F\Box p$	(24)[19]
1121	$F\Diamond\Box p$	(25)[19]

1122	Fp	(26)[19]
1122	$T\Diamond p$	(27)[18]
1122	$T\Box\Diamond p$	(28)[18]
1122	$F\Box p$	(29)[19]
1122	$F\Diamond\Box p$	(30)[19]

Zu 3.8.1 Dynamische Logik

zu Aufgabe 3.8.2

zu Aufgabe 3.8.2

zu Aufgabe 3.8.3 Die Strukturen $\mathcal{A}_n = (W_n, \tau_n, \rho_n)$ seien für $n \geq 1$ gegeben durch $W_n = \{w_1, \ldots, w_n\}$ und $\rho_n(a) = \{(w_i, w_{i+1}) \mid 1 \leq i < n\}$. Wir nehmen an, daß außer a kein weiteres atomare Programm vorkommt. τ spielt eigentlich keine tragende Rolle und wir setzen für alle n und alle aussagenlogische Variablen p $\tau_n(p) = W_n$. Schließlich definieren wir noch die unendliche Strukture $\mathcal{A}_\omega = (W_\omega, \tau_\omega, \rho_\omega)$ durch $W_\omega = \{w_i \mid 1 \leq i\} = \bigcup_{i \geq 1} W_n$, $\rho_\omega(a) = \{(w_i, w_{i+1}) \mid 1 \leq i\} = \bigcup_{i \geq 1} \rho_n(n)$ und $\tau_\omega(p) = W_\omega$ für alle n.

Angenommen es gäbe eine Formel $F \in Fml_{PDLmin}$ die

> es gibt keine unendliche Folge $w_0, w_1, \ldots$ von Zusänden $w_i \in W$ mit $(w_i, w_{i+1}) \in \rho^{\mathcal{A}}(a)$ für alle $i \geq 0$.

ausdrückt, dann gilt
$(\mathcal{A}_n, w_1) \models F$ für alle $n \geq 1$ aber
$(\mathcal{A}_\omega, w_1) \models \neg F$

Wir zeigen und daraus ergibt sich dann ein Widerspruch:

1. für alle Programme α aus Π_{min} gilt
   (a) $\rho_1(\alpha) \subseteq \rho_1(\alpha) \subseteq \ldots \rho_n(\alpha) \subseteq \ldots$
   (b) $\rho_\omega(\alpha) = \bigcup_{n \geq 1} \rho_n(\alpha)$
2. für jede Formel $G \in Fml_{PDLmin}$ und jedes i gilt

$$(\mathcal{A}_\omega, w_i) \models G \text{ gwd } (\mathcal{A}_n, w_i) \models G \text{ für unendlich viele } n \geq i$$

Die Beweise erfolgen durch simultane Induktion über die Komplexität von Programmen und Formeln.

Beweis von (1) Ist α atomar, dann ist $\alpha = a$ und die Behauptung folgt aus der Definition von $\mathcal{A}_\omega$. Sei jetzt $\alpha = \alpha_1; \alpha_2$. Nach Induktionsvoraussetzung können wir

Monotonie von $(\rho_n(\alpha_i))_{n \geq 1}$ und $\rho_\omega(\alpha_i) = \bigcup_{n \geq 1} \rho_n(\alpha_i) \quad$ für $i = 1, 2$

annehmen.

$$\begin{array}{rcl}
\rho_\omega(\alpha_1;\alpha_2) & = & \{(s_1,s_2) \mid \text{ es gibt } s \text{ mit } (s_1,s) \in \rho_\omega(\alpha_1) \text{ und } (s,s_2) \in \rho_\omega(\alpha_2)\} \\
& = & \{(s_1,s_2) \mid \text{ es gibt } s \text{ mit} \\
& & \quad (s_1,s) \in \bigcup_{n\geq 1} \rho_n(\alpha_1) \text{ und } (s,s_2) \in \bigcup_{n\geq 1} \rho_n(\alpha_2)\} \\
& = & \{(s_1,s_2) \mid \text{ es gibt } s \text{ und } n \text{ mit} \\
& & \quad (s_1,s) \in \rho_n(\alpha_1) \text{ und } (s,s_2) \in \rho_n(\alpha_2)\} \\
& = & \{(s_1,s_2) \mid \text{ es gibt } n \text{ mit } (s_1,s_2) \in \rho_n(\alpha_1;\alpha_2)\} \\
& = & \bigcup_{n\geq 1} \rho_n(\alpha_1;\alpha_2)
\end{array}$$

Die Monotonie von $(\rho_n(\alpha_1;\alpha_2))_{n\geq 1}$ ist offensichtlich.

$$\begin{array}{rcl}
\rho_\omega(\alpha^*) & = & \{(s_1,s_2) \mid \text{ es gibt } k \text{ und } t_1,\dots t_k \text{ mit } t_1 = s_1, t_k = s_2 \\
& & \quad \text{und } (t_j,t_{j+1}) \in \rho_\omega(\alpha) \text{ für } 1 \leq j < k\} \\
& = & \{(s_1,s_2) \mid \text{ es gibt } k \text{ und } t_1,\dots t_k \text{ mit } t_1 = s_1, t_k = s_2 \\
& & \quad \text{und } (t_j,t_{j+1}) \in \bigcup_{n\geq 1} \rho_n(\alpha) \text{ für } 1 \leq j < k\} \\
& = & \{(s_1,s_2) \mid \text{ es gibt } k,\ t_1,\dots t_k \text{ und } n \text{ mit } t_1 = s_1, t_k = s_2 \\
& & \quad \text{und } (t_j,t_{j+1}) \in \rho_n(\alpha) \text{ für } 1 \leq j < k\} \\
& = & \{(s_1,s_2) \mid \text{ es gibt } n \text{ mit } (s_1,s_2) \in \rho_n(\alpha^*)\} \\
& = & \bigcup_{n\geq 1} \rho_n(\alpha^*)
\end{array}$$

Die Monotonie von $(\rho_n(\alpha^*))_{n\geq 1}$ ist wieder offensichtlich.

$$\begin{array}{rcl}
\rho_\omega(?G) & = & \{(s,s) \mid (\mathcal{A}_\omega,s) \models G\} \\
& = & \{(s,s) \mid \text{ es gibt } n \text{ mit } (\mathcal{A}_i,s) \models G \\
& & \quad \text{für unendlich viele } i \geq n\} \\
& = & \{(s,s) \mid \text{ es gibt } n \text{ mit } (s,s) \in \rho_i(?G) \\
& & \quad \text{für unendlich viele } i \geq n\} \\
& = & \bigcup_{n\geq 1} \rho_n(?G)
\end{array}$$

Die Monotonie von $(\rho_n(?G))_{n\geq 1}$ ist wieder einfach einzusehen.

Beweis von (2) Ist $G = p$ eine aussagenlogische Variable dann ist die Behauptung offensichtlich wegen der trivialen Definition von τ. Der Fall $G = G_1 \wedge G_2$ bereitet ebenfalls keine Schwierigkeiten.

$$\begin{array}{rcl}
(\mathcal{A}_\omega,w_i) \models G_1 \vee G_2 & \Leftrightarrow & (\mathcal{A}_\omega,w_i) \models G_1 \text{ oder } (\mathcal{A}_\omega,w_i) \models G_2 \\
& \Leftrightarrow & (\mathcal{A}_n,w_i) \models G_1 \text{ für unendlich viele } n \geq i \\
& & \ \text{oder} \\
& & (\mathcal{A}_n,w_i) \models G_2 \text{ für unendlich viele } n \geq i \\
& \Leftrightarrow & (\mathcal{A}_n,w_i) \models G_1 \vee G_2 \text{ für unendlich viele } n \geq i
\end{array}$$

$$
\begin{array}{rcl}
(\mathcal{A}_\omega, w_i) \models [\alpha]G & \Leftrightarrow & \text{für alle } (w_i, w_j) \in \rho_\omega(\alpha) \text{ gilt } (\mathcal{A}_\omega, w_j) \models G \\
& \Leftrightarrow & \text{für alle } j \text{ mit } (w_i, w_j) \in \bigcup_{n\geq 1} \rho_n(\alpha) \\
& & \quad (\mathcal{A}_n, w_j) \models G \text{ für unendlich viele } n \geq j \\
& \Leftrightarrow & \text{für alle } j, n \text{ mit } (w_i, w_j) \in \rho_n(\alpha) \\
& & \quad (\mathcal{A}_n, w_j) \models G \text{ für unendlich viele } n \geq j \\
& \Leftrightarrow & (\mathcal{A}_n, w_j) \models [\alpha]G \text{ für unendlich viele } n \geq j
\end{array}
$$

Im letzten Beweisschritt habe wir in der Richtung von oben nach unten Gebrauch gemacht von der Endlichkeit von $\rho_n(\alpha)$. Der Induktionsschritt für $\langle\alpha\rangle G$ geschieht analog.

■

zu Aufgabe 3.8.4 Wir versuchen den Beweisweg von Satz 81 auf PDL^- zu erweitern. Als erstes muß die Definition 90 des Fischer-Ladner Abschlusses erweitert werden durch:

$$< \alpha^- > F \in S \Rightarrow < \alpha > F \in S$$

Die Definitionen der Äquivalenzrelation $\sim_S$ (Def. 91)und der Qutientenstruktur $\mathcal{A}_S$ (Def.92) können unverändert übernommen werden. Es bleibt die Erweiterung von Lemma 80 nachzuweisen. Diese Aufgabe reduziert sich auf den Beweis des zusätzlichen Falls im Induktionsschritt:

$$\text{aus } (w_1, w_2) \in \rho(\alpha^-) \text{ folgt } ([w_1], [w_2]) \in \rho_S(\alpha^-)$$

Das ist aber offensichtlich:

$$
\begin{array}{rcll}
(w_1, w_2) \in \rho(\alpha^-) & \Leftrightarrow & (w_2, w_1) \in \rho(\alpha) & \text{Def. von } \alpha^- \\
& \Rightarrow & ([w_2], [w_1]) \in \rho_S(\alpha) & \text{Ind.vor.} \\
& \Rightarrow & ([w_1], [w_2]) \in \rho_S(\alpha^-) & \text{Def. von } \alpha^-
\end{array}
$$

■

Zu 4.1 Temporale Logik

zu Aufgabe 4.1.1 Die einzige Stelle, an der die Linearität der Ordnungen in $\mathcal{M}$ benutzt wird, ist beim Übergang von

$$(\mathcal{H}, s) \models C_j \text{ gdw } \mathcal{H}_{temp} \models G_j^0(s)$$

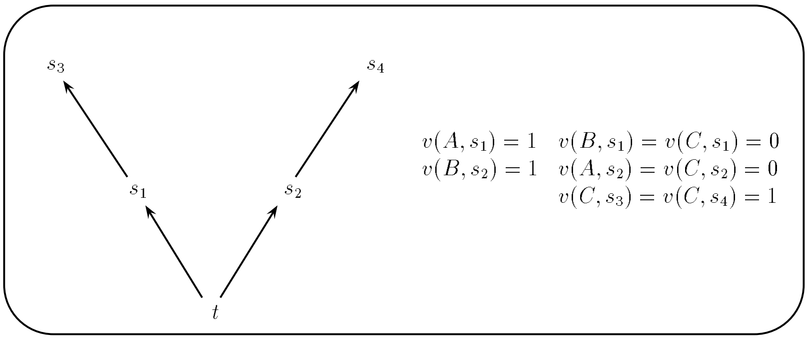


Abbildung 5.2: Gegenbeispiel für Aufgabe 4.1.2

zu

$$(\mathcal{H}, t) \models sif\ C_j \vee C_j \vee sip\ C_j \text{ gdw } \mathcal{H}_{temp} \models \exists y G_j^0(t, y)$$

Hier zu muß für jede TL-Formel A, jede Struktur $\mathcal{K} = (G, v, <)$ mit $(G, <) \in \mathcal{M}$ und jedes $t \in G$ die Aussage

„es gibt ein $s \in G$ mit $(\mathcal{K}, s) \models A$“

äquivalent sein zu

„$(\mathcal{K}, t) \models sip\ A \vee A \vee sif\ A$“.

zu Aufgabe 4.1.2

1. Die folgenden Zeilen sind jeweils zueinander äquivalent:
   - $(\mathcal{K}, t) \models C\ until\ (A \vee B)$
   - $\exists s > t(\mathcal{K}, s) \models (A \vee B)$ und $\forall s'(s > s' > t \Rightarrow (\mathcal{K}, s') \models C$
   - ($\exists s > t(\mathcal{K}, s) \models A$ oder $\exists s > t(\mathcal{K}, s) \models B$) und $\forall s'(s > s' > t \Rightarrow (\mathcal{K}, s') \models C$
   - ($\exists s > t(\mathcal{K}, s) \models A$ und $\forall s'(s > s' > t \Rightarrow (\mathcal{K}, s') \models C$) oder ($\exists s > t(\mathcal{K}, s) \models B$ und $\forall s'(s > s' > t \Rightarrow (\mathcal{K}, s') \models C$)
   - $(C\ until\ A)$ oder $(C\ until\ B)$
   - $(C\ until\ A) \vee (C\ until\ B)$
2. Für die Zeitstruktur $\mathcal{K}$ aus Abbildung 5.2 gilt $(\mathcal{K}, t) \models (A\ until\ C)$ und $(\mathcal{K}, t) \models (B\ until\ C)$, aber nicht $(\mathcal{K}, t) \models (A \wedge B)\ until\ C$.

zu Aufgabe 4.1.3 $A_{alt} = p \wedge \Box(p \to \bigcirc\neg p) \wedge \Box(\neg p \to \bigcirc p)$. Definition von $\Box$ und $\bigcirc$ siehe Seite 243.

Zu 4.2 Verzweigende Temporale Logiken

zu Aufgabe 4.2.3
Zu 1: Sei $G = \{n \in \mathbb{N} \mid n \leq a\}$ für $a > 0$ ein endliches Anfangsstück der natürlichen Zahlen und f definiert durch

$$f(X) = \{n + 1 \mid n \in X \text{ und } n < a\}$$

Es gilt, z.B. $f(\emptyset) = \emptyset$ und $f(G) = \{n \mid 1 \leq n \leq a\}$. Sicherlich ist f eine monotone Funktion und man sieht leicht, daß $\emptyset$ der keinste und auch der größste, und damit der einzige, Fixpunkt ist. Für keine nichtleere Menge M gibt es also einen Fixpunkte Z von f mit $M \subseteq Z$.
Zu 2: Erfüllt die Menge M die Inklusionsbeziehung $M \subseteq f(M)$, dann kann der Beweis von Lemma 103 praktisch unverändert übernommen werden um die Existenz eines kleinsten Fixpunkts oberhalb von M zu beweisen.

zu Aufgabe 4.2.4 Als Vorüberlegung zeigt man zunächst die Monotonie von f_{AX}, was leicht einzusehen ist.
Die Behauptung $f^n(\emptyset) \subseteq f^{n+1}(\emptyset)$ wird jetzt durch Induktion nach n bewiesen.
Für $n = 0$ ergibt sich sofort $\emptyset = f^0(\emptyset) \subseteq f^1(\emptyset)$.
Setzen wir jetzt für $n \geq 1$ $f^{n-1}(\emptyset) \subseteq f^n(\emptyset)$ voraus und versuchen $f^n(\emptyset) \subseteq f^{n+1}(\emptyset)$ zu beweisen. Sei also $g \in f^n(\emptyset)$ Nach Definition von f folgt daraus $g \in \tau(F_2) \cup (\tau(F_1) \cap f_{AX}(f^{n-1}(\emptyset))$ Nach der einleitended bemerkten Monotonie von f_{AX} und der Induktionsannahme folgt $f_{AX}(f^{n-1}(\emptyset) \subseteq f_{AX}(f^n(\emptyset))$ und damit auch $g \in f^{n+1}(\emptyset)$

Kapitel 6

Appendix: Programs

6.1 ReverseList

```
import java.io.*;
class ReverseList {
    int value; ReverseList next;
    ReverseList(int val,ReverseList n){value = val ; next = n;}

    public static void main (String[] args) {
        int l,in,c; ReverseList h=null,h0,r;
        System.out.print("Length of list? "); l = LiesInt();
        System.out.println();
        if (l!=0) {
        System.out.print("Enter value for first cell ");
        in = LiesInt();h = new ReverseList(in,null);
        l = l-1;}
        while (l!=0) {
         if (l==1) {System.out.print("Enter value for last cell ");
             in = LiesInt();h = new ReverseList(in,h);
             l = l-1; }
         else {System.out.print("Enter value for next cell ");
               in = LiesInt();h = new ReverseList(in,h);
               l = l-1;}}
        h.printList(); h.reverse().printList();
        System.out.println();}

    public ReverseList reverse() {
        ReverseList t, y= null, x = this;
        while (x != null) {t=y;y=x;x=x.next;y.next = t;}
        return y;}

    public void printList(){
      ReverseList ll = this;
      System.out.println();
      while (ll != null){System.out.print(ll.value +" ");ll = ll.next;}}

 static int LiesInt() {
   DataInput StdEingabe = new DataInputStream(System.in);
   int ergebnis = 0;
   try{ ergebnis = Integer.parseInt(StdEingabe.readLine()); }
   catch (IOException io) {} return ergebnis;}}
```

Literaturverzeichnis

[Ackermann, 1935] Wilhelm Ackermann. Untersuchungen über das eliminationsproblem der mathematischen logik. *Mathematische Annalen*, 110:390–413, 1935.

[Asser, 1959] G. Asser. *Einführung in die mathematische Logik. Teil I: Aussagenkalkül.* Teubner Verlag, Leipzig, 6. edition, 1959.

[Baader *et al.*, 2003] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, & Peter Patel-Schneider, editors. *The Description Logic Handbook.* Cambridge University Press, 2003.

[Bernays, 1926] P. Bernays. Untersuchungen des Aussagenkalküls der *principia mathematica*. *Mathematische Zeitschrift*, 25:305–320, 1926.

[Bicarregui *et al.*, 1994] Juan C. Bicarregui, John S. Fitzgerald, Peter A. Lindsay, Richard Moore, & Brian Ritchie. *Proof in VDM: A Practitioner's Guide.* FACIT Formal Approaches to Computing and Information Technology. Springer Verlag, 1994.

[Blamey, 1986] S. Blamey. Partial logic. In F. Guenthner D. Gabbay, editor, *Handbook of Philosophical Logic*, volume III, Alternatives in Classical Logic, pages –. D. Reidel, Synthese Library, 1986.

[Bolc & Borowik, 1992] Leonard Bolc & Piotr Borowik. *Many-Valued Logics*, volume 1. Springer Verlag, 1992.

[Bolc & Szałas, 1995] Leonard Bolc & Adrzej Szałas, editors. *Time & Logic. A computational approach.* University College London Press, 1995.

[Brachmann & Schmolze, 1985] R. J. Brachmann & J. G. Schmolze. An overview of the kl-one knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[Brachmann, 1977] R. J. Brachmann. What's in a concept: Structural foundations for semantic networks. *International J. of Man-Machine Studies*, 9:127–152, 1977.

[Brachmann, 1978] R. J. Brachmann. Structured inheritance networks. In *Research in Natural Language Understanding*, Quarterly Progress Report No.1, BBN Report No. 3742, pages 36–78, Cambridge, Mass., 1978. Bolt, Berank and Newman Inc.

[Breuer, 1972] M. A. Breuer. A note on three-valued logic simulation. *IEEE Transactions on Computers*, C21(4):399–402, April 1972.

[Bull & Segerberg, 1984] Robert A. Bull & Krister Segerberg. Basic modal logic. In *Handbook of Philosophical Logic*, volume II Extensions of Classical Logic, pages 1 – 88. D.Reidel, 1984.

[Carnielli, 1987] Walter A. Carnielli. Systematization of finite many-valued logics through the method of tableaux. *Journal of Symbolic Logic*, 52(2):473–493, 1987.

[Carnielli, 1991] W. A. Carnielli. On sequents and tableaux for many-valued logics. *Journal of Non-Classical Logic*, 8(1):59–76, 1991.

[Clarke & Sistla, 1985] E.M. Clarke & A.P. Sistla. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32:733–749, 1985.

[Clarke *et al.*, 1986] E.M. Clarke, E.A. Emerson, & A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[Clarke *et al.*, 1993] E. M. Clarke, O. Grumberg, & D. Long. Verification tools for finite state concurrent systems. In de Bakker et al. [de Bakker *et al.*, 1993], pages 124 – 175.

[Clarke *et al.*, 2001] Edmund M. Clarke, Rna Grumberg, & Doron A. Peled. *Model Checking*. The MIT Press, 2001.

[de Bakker *et al.*, 1993] J.W. de Bakker, W.P. de Roever, & G. Rozenberg, editors. *A Decade of Concurrency - Reflections and Perspectives*, volume 803 of *Lecture Notes in Computer Science*. Springer-Verlag, June 1993.

[Doherty, 1996] Patrick Doherty, editor. *Partiality, Modality and Nonmonotonicity.* CLSI Publications, Stanford, California, 1996.

[Eichelberger, 1965] E. B. Eichelberger. Hazard detection in combinational and sequential switching circuits. *IBM J.Res.Develop.*, 9:90–99, March 1965.

[Emerson, 1992] E. Allen Emerson. Temporal and modal logic. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science. Volume B. Formal Models and Semantics*, pages 996–1072. Elevesier, 1992.

[Epstein, 1993] George Epstein. *Multiple-valued logic design : an introduction.* Institute of Physics Publishing, Bristol, England; Philadelphia, 1993.

[Fenstad *et al.*, 1987] J.E. Fenstad, P.K. HJalvorsen, T. Langholm, & J van Benthem. *Situations, Language and Logic.* Reidel, Dortrecht, 1987.

[Fitting, 1983] Melvin Fitting. *Proof Methods for Modal and Intuitionistic Logic.* D. Reidel, 1983.

[Fitting, 1991] Melvin C. Fitting. Modal logic should say more than it does. In Jean-Louis Lassez & Gordon Plotkin, editors, *Computational Logic, Essays in Honor of Alan Robinsion*, pages 113 – 135. MIT Press, Cambridge, MA, 1991.

[Fitting, 1993] Melvin Fitting. Basic modal logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume Vol. 1 Logical Foundations, pages 368 – 448. Clarendon Press, 1993.

[Franconi *et al.*, 2000] Enrico Franconi, Franz Baader, Ulrike Sattler, & Panos Vassiliadis. Multidimensional data models and aggregation. In Jarke et al. [Jarke *et al.*, 2000], pages 86–105, Chapter 5.

[Fujiwara, 1986] Hideo Fujiwara. *LOGIC TESTING AND DESIGN FOR TESTABILITY.* SERIES IN COMPUTER SCIENCE. MIT PRESS, Cambridge Massachusetts, 1986.

[Gabbay *et al.*, 1994] Dov M. Gabbay, Ian Hodkinson, & Mark Reynolds. *Temporal Logic. Mathematical Foundations and Computational Aspects*, volume 1 of *Oxford Logic Guides, Vol. 28*. Clarendon Press, Oxford, 1994.

[Gabbay, 1994] Dov M. Gabbay. Classical vs. non-clasical logics (the universality of classical logic). In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume Vol. 2 Deduction Methodologies, pages 359 – 500. Clarendon Press, 1994.

[Garson, 1984] James W. Garson. Quantification in modal logic. In *Handbook of Philosophical Logic*, volume II Extensions of Classical Logic, pages 249 – 308. D.Reidel, 1984.

[Goldblatt, 1982] Robert Goldblatt. *Axiomatising the logic of computer programming*, volume 130 of *LNCS*. Springer-Verlag, 1982.

[Goldblatt, 1987] Robert Goldblatt. *Logics of Time and Computation.* Number 7 in CSLI Lecture Notes. CSLI, 1987.

[Gottwald, 1990] Siegfried Gottwald. *Nichtklassische Logik*, chapter 1, pages 19 – 85. Akademie-Verlag, Berlin, 1990.

[Gottwald, 1993] Siegfried Gottwald. *Fuzzy Sets and Fuzzy Logic.* Vieweg, 1993.

[Gottwald, 2001] Siegfried Gottwald. *A Treatise On Many-Valued Logics*, volume 9 of *Studies in Logic and Computation.* Research Studies Press, Baldock, Hertfordshire, England, 2001.

[Grädel & Otto, 1999] E. Grädel & M. Otto. On Logics with Two Variables. *Theoretical Computer Science*, 224:73–113, 1999.

[Gries, 1989] David Gries. *The Science of Programming.* Texts and Monographs in Computer Science. Springer Verlag, 1989.

[Grädel, 1999] Erich Grädel. Why are modal logics so robustly decidable? *Bulletin of EATCS*, 68:90–103, 1999.

[Hähnle, 1993] Reiner Hähnle. *Automated Deduction in Multiple-valued Logics.* Clarendon Press, Oxford, 1993.

[Hájek, 1998] Petr Hájek. *Metamathematics of Fuzzy Logic*, volume 4 of *Trends in Logic - Studia Logica Library*. Kluwer Academic Publishers, 1998.

[Harel *et al.*, 2000] David Harel, Dexter Kozen, & Jerzy Tiuryn. *Dynamic Logic*. The MIT Press, 2000.

[Harel, 1979] David Harel. *First-Order Dynamic Logic*, volume 68 of *Lecture Notes in Computer Science*. Springer-Verlag, 1979.

[Harel, 1984] David Harel. Dynamic logic. In *Handbook of Philosophical Logic*, volume II Extensions of Classical Logic, pages 497 – 604. D.Reidel, 1984.

[Hayes, 1986a] J. P. Hayes. Digital simulation with multiple logic values. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, CAD-5(2):274–283, April 1986.

[Hayes, 1986b] J. P. Hayes. Uncertainty, energy , and multiple logic values. *IEEE Trans. on Computers*, C-35(2):107–114, February 1986.

[Heuerding, 1998] Alain Heuerding. *Sequent Caclculi for Proof Search in Some Modal Logics. Theory - Implementation - Application.* PhD thesis, Universität Bern, Philisophisch-naturwissenschaftliche Fakultät, 1998.

[Holzmann, 1991] Gerard J. Holzmann. *Design and validation of computer protocols*. Prentice Hall, 1991.

[Horrocks *et al.*, 2000] I. Horrocks, U. Sattler, & S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.

[Hughes & Cresswell, 1972] G. E. Hughes & M.J. Cresswell. *An Introduction to Modal Logic*. Methuen and Co Ltd, London, second edition, 1972.

[Hughes & Cresswell, 1984] G. E. Hughes & M.J. Cresswell. *A Companion to Modal Logic*. Methuen and Co Ltd, London, 1984.

[J. B. Rosser, 1952] A. R. Turquette J. B. Rosser. *Many-valued Logics*. North-Holland, 1952.

[J. Los, 1958] R. Saszko J. Los. Remarks on sentential logics. *Indagationes Mathematicae*, 20:177–183, 1958.

[Jarke *et al.*, 2000] Matthias Jarke, Maurizio Lenzerini, Yannis Vassiliou, & Panos Vassiliadis, editors. *Fundamentals of Data Warehouses*. psv, 2000.

[Jephson *et al.*, 1969] J. S. Jephson, R. P. McQuarrie, & R. E. Vogelsberg. A three-value computer design verification system. *IBM J.Res.Develop.*, 8(3):178–188, 1969.

[Jones, 1990] Cliff B. Jones. *Systematic Software Development using VDM*. International Series in Computer Science. Prentice Hall, 2. edition, 1990.

[Kamp, 1968] Hans Kamp. *Tense Logic and the theory of Linear Order*. PhD thesis, Department of Philosophy, U.C.L.A., 1968.

[Kleene, 1938] S.C. Kleene. On a notation for ordinal numbers. *JournaL of Symbolic Logic*, 3:150–155, 1938.

[Lamport, 1985] Lesli Lamport. The mutual exclusion problem–part i: A theory of interprocess communication, part ii: Statement and solutions. *Journal of the ACM*, 33(2):313–348, January 1985.

[Langholm, 1996] Tore Langholm. How different is partial logic? In Doherty [Doherty, 1996], pages 3–43.

[Lewis, 1918] C. I. Lewis. *A survey of symbolic logic*. University of California, Berkeley, 1918.

[Łukasiewicz, 1920] Jan Łukasiewicz. O logice tr ojwartościowej (on 3-valued logic). *Ruch Filozoficzny*, 5:169–171, 1920.

[Łukasiewicz, 1930] Jan Łukasiewicz. Philosophische Bemerkungen zu mehrwertigen Systemen des Aussagenkalküls. *Comptes rendus des séance de la Societé des Sciences et des Lettres de Varsovie, Classe III*, 23:51–77, 1930.

[Malinowski, 1993] Grzegorz Malinowski. *Many-Valued Logics*. Oxford Logic Guides, Vol. 25. Clarendon Press, Oxford, 1993.

[Marx & Venema, 1997] Maarten Marx & Yde Venema. *Multi-Dimensional Modal Logic*, volume 4 of *Applied Logic Series*. Kluwer Academic Publishers, Dordrecht,Boston, London, 1997.

[Max, 1990] Ingolf Max. Präsuppositionen - ein Überblick über die logischen Darstellungsweisen. In Lothar Kreiser, Siegfried Gottwald, & Werner Stelzner, editors, *Nichtklassische Logik, Eine Einführung*, pages 349 –406. Akademie-Verlag, Berlin, 1990.

[McNaughton, 1951] R. McNaughton. A theorem about infinite-valued Łukasiewicz calculi. *The Journal of Symbolic Logic*, 16:1 – 13, 1951.

[M.Gabbay & Guenthner, 2001] Dov M.Gabbay & Franz Guenthner, editors. *Handbook of Philosophical Logic. 2nd Edition. Volume 3.* Kluwer Academic Publishers, 2001.

[Mortimer, 1975] M. Mortimer. On languages with two variables. *Zeitschrift für mathematische Logik und Grundlagen*, 21:135–140, 1975.

[Muller, 1959] D. E. Muller. Treatment of transition signals in electronic switching circuits by algebraic methods. *IRE Trans. on Electronic Computers*, EC-8:401ff, 1959.

[Mundici, 1994] Daniele Mundici. A constructive proof of McNaugthon's Theorem in infinite-valued logics. *The Journal of Symbolic Logic*, 59:569–602, 1994.

[Murray, 1999] N. V. Murray, editor. *Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux'99*, number 1617 in Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, June 1999.

[Muth, 1976] P. Muth. A nine-valued circuit model for test generation. *IEEE Trans.Cpmuter*, C-25(6):630–636, 1976.

[Muzio & Wesselkamper, 1986] J.C. Muzio & T.C. Wesselkamper. *Multiple-valued switching theory.* Adam Hilger Ltd, Bristol and Boston, 1986.

[Nadolny, 1987] Sten Nadolny. *Die Entdeckung der Langsamkeit.* Piper, München, Zürich, 1987.

[Nielson *et al.*, 1999] Flemming Nielson, Hanne Riis Nielson, & Chris Hankin. *Principles of Program Analysis.* Springer, 1999.

[Patel-Schneider & Horrocks, 1999] P. F. Patel-Schneider & I. Horrocks. DLP and FaCT. In Murray [Murray, 1999], pages 19–23.

[Pomeranz & Reddy, 1995] I. Pomeranz & S. M. Reddy. On fault simulation for synchronous sequential circuits. *IEEE Transactions on Computers*, 44(2):335–340, February 1995.

[Post, 1921] E. Post. Introduction to a general theory of elementary propositions. *American J. of Math.*, 43:163–185, 1921.

[Rescher, 1989] N. Rescher. *Many-valued Logic.* McGraw-Hill, 1989.

[Rosenberg, 1984] Ivo G. Rosenberg. Completeness properties of multiple-valued logic algebras. In D. C. Rine, editor, *Computer Science and Multiple-Valued Logic*, pages 150–192. North-Holland, 1984.

[Sagiv *et al.*, 1999] Mooly Sagiv, Thomas Reps, & Reinhard Wilhelm. Parametric shape analysis via 3-valued logic. In *Proceedings of the 26th ACM Symposium on Principles of Programming Languages*, pages –, January 1999.

[Sagiv *et al.*, 2000] M. Sagiv, T. Reps, & R. Wilhelm. Parametric shape analysis via 3-valued logic. Tech report TR-1383, Computer Science Department, Univ. of Wisconsin, Madison, WI, March 2000. revised version.

[Sagiv *et al.*, 2001] Moody Sagiv, Thomas Reps, & Reinhard Wilhelm. Parametric shape analysis via 3-valued logic. *to appear*, 2001.

[Sahlqvist, 1975] H. Sahlqvist. Completeness and correspondance in first and second order semantics for modal logic. In S. Kanger, editor, *Proceedings of the Third Scandinavian Logic Colloquium*, pages 110–143. North Holland, 1975.

[Schild, 1991] Klaus Schild. A correspondence theory for terminological logics: preliminary report. In *Proc.of the 12.Int.Joint Conf. on Artifical Intelligence (IJCAI'91)*, pages 466–471, 1991.

[Schild, 1993] Klaus Schild. Combining terminological logics with tense logic. In *Proc. 6th Portuguese Conference on Artificial*

Intelligence, EPIA-93, volume 727 of *Lecture Notes in Computer Science*, pages 105–120. Springer, 1993.

[Słupecki, 1972] J. Słupecki. Completeness criterion for many-valued propositional caculus. *Studia Logica*, 30:153–157, 1972.

[S.McCall, 1967] S.McCall. *Polish Logic: 1920 - 1939.* Oxford University Press, 1967.

[Smith & Glulak, 1993] K. C. Smith & P. G. Glulak. Prospects for multiple-valued integrated circuits. *IEICE Trans. Electron.*, E76-C(3), march 1993.

[Smullyan, 1968] Raymond Smullyan. *First-Order Logic*. Springer, 1968.

[Steinacker, 1990] Peter Steinacker. *Nichtklassische Logik*, chapter 3, pages 86 – 159. Akademie-Verlag, Berlin, 1990.

[Sterling, 1992] Colin Sterling. Modal and temporal logics. In Dov. M. Gabbay S. Abramsky & T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume Vol. 2 Background: Computational Structures, pages 478 – 563. Clarendon Press, 1992.

[Surma, 1984] S. J. Surma. An algorithm for axiomatizing every finite logic. In D. C. Rine, editor, *Computer Science and Multiple-Valued Logic*, pages 143–149. North-Holland, 1984.

[Szałas, 1993] Andrzej Szałas. On the correspondence between modal and classical logic: an automated approach. *J. Logic and Computation*, 3:605–620, 1993.

[Tarski, 1930] A. Tarski. Fundamentale Begriffe der Methodologie der deduktiven Wissenschaft I. *Monatshefte Mathematik und Physik*, pages 361–404, 1930.

[Tarski, 1935] A. Tarski. Grundzüge des Systemkalküls I. *Fundamentae Mathematicae*, 25:503–526, 1935.

[Urquhart, 1986] A. Urquhart. Many-valued logic. In F. Guenthner D. Gabbay, editor, *Handbook of Philosophical Logic*, volume III, pages 70–116. D. Reidel, Synthese Library, 1986.

[van Benthem, 1984] Johan van Benthem. Correspondence theory. In *Handbook of Philosophical Logic*, volume II Extensions of Classical Logic, pages 167 – 248. D.Reidel, 1984.

[van Benthem, 2001] Johan van Benthem. Handbook of philosophical logic. 2nd edition. volume 3. In M.Gabbay & Guenthner [M.Gabbay & Guenthner, 2001], pages 325–408.

[V.R.Pratt, 1976] V.R.Pratt. Semantical considerations on Floyd-Hoare logic. In *Proceedings of the 17 th IEEE Symp. Foundations of Computer Science*, pages 109 – 121. IEEE, 1976.

[Wajsberg, 1931] M. Wajsberg. Aksjomatyzacja trójwartościowego rachunku zdań. *Comptes Rendus de la Socieété des Sciences et des Lettres de Varsovic, Cl. III*, 24:126–148, 1931.

[Wolper, 1983] Pierre Wolper. Temporal logic can be more expressive. *Information and Control*, 56(1–2):72–99, 1983.

[Wolper, 1995] Pierre Wolper. On the relation of programs and computations to models of temporal logic. In Bolc & Szałas [Bolc & Szałas, 1995], pages 131–178, Chapter 3.

[Yoeli & Rinon, 1964] M. Yoeli & S. Rinon. Applications of ternary algebra to the study of static hasards. *J.ACM*, 11(84), 1964.

Index