Prädikatenlogische Kalküle

Resolutionskalkül

Syntax:

Definition

- Ein **Literal** ist eine atomare oder eine negierte atomare Formel.
- Eine **Klausel** ist eine endliche Menge von Literalen.
- Die **leere Klausel** wird mit $\square$ bezeichnet.
- Eine Klausel wird interpretiert wie die Disjunktion ihrer Literale,
- Eine Menge von Klauseln wird interpretiert wie die Konjunktion ihrer Klauseln

Notation:

Zu einem Literal L sei $\sim L$ das Literal

$$\sim L := \begin{cases} \neg L & \text{wenn } L \text{ ein Atom ist} \\ L' & \text{wenn } L = \neg L' \end{cases}$$

Zu einer Klausel C sei

$$\sim C := \{\sim L \mid L \in C\}.$$

Resolution, spezieller Fall:

- C_1, C_2 Klauseln $p(t_1)$, $\neg p(t_2)$ Literale
- $Var(C_1 \cup \{p(t_1)\}) \cap Var(C_2 \cup \{p(t_2)\}) = \emptyset$
- μ ist allgemeinster Unifikator von $p(t_1)$ und $p(t_2)$.

$$\frac{C_1 \cup \{p(t_1)\} \qquad C_2 \cup \{\neg p(t_2)\}}{\mu(C_1 \cup C_2)}$$

Die Klausel $\mu(C_1 \cup C_2)$ heißt eine *Resolvente* der Eingabeklauseln $C_1 \cup \{p(t_1)\}$ und $C_2 \cup \{\neg p(t_2)\}$.

Die Resolutionsregel (allgemeiner Fall):

- C_1, C_2, K_1, K_2 sind Klauseln
- K_1, $K_2 \neq \Box$
- $Var(C_1 \cup K_1) \cap Var(C_2 \cup K_2) = \emptyset$
- μ ist allgemeinster Unifikator von $K_1 \cup \sim K_2$.

$$\frac{C_1 \cup K_1 \qquad C_2 \cup K_2}{\mu(C_1 \cup C_2)}$$

Die Klausel $\mu(C_1 \cup C_2)$ heißt eine *Resolvente* der Eingabeklauseln $C_1 \cup K_1$ und $C_2 \cup K_2$.

Korrektheit und Vollständigkeit:

Sei K eine Menge von Klauseln im Vokabular Σ. Dann gilt

1. Gilt $M \vdash_{\mathbf{R}} \Box$, dann ist M nicht erfüllbar.
2. Ist M nicht erfüllbar, dann folgt $M \vdash_{\mathbf{R}} \Box$

Beweis: s. Schmitt (2008), S. 180ff.

Anwendung der Resolutionsregel (Beispiel):

Regelschema:

$$\frac{C_1 \cup K_1 \qquad C_2 \cup K_2}{\mu(C_1 \cup C_2)}$$

Gegeben seien die beiden Klauseln
$\{p(x), q(f(x)), q(f(g(c))\}$ und $\{r(y,z), \neg q(z), \neg q(f(y))\}$

$$\frac{\{p(x)\} \cup \{q(f(x)), q(f(g(c)))\} \qquad \{r(y,z)\} \cup \{\neg q(z), \neg q(f(y))\}}{\mu(C_1 \cup C_2)}$$

$K_1 \cup \sim K_2$ ist in diesem Fall $\{q(z), q(f(y)), q(f(x)), q(f(g(c))\}$
Der allgemeinste Unifikator ist
$\mu = \{x/g(c),\ y/g(c),\ z/f(g(c))\}$
Die Resolvente ist

$$\{p(g(c)), r(g(c), f(g(c)))\}$$

Anwendung des Resolutionskalküls

Sei M eine Klauselmenge.

1. Mit $Res(M)$ bezeichen wir die Menge aller Resolventen von Klauseln aus M. Genauer:

$$\begin{aligned} Res(M) = \ & \{B \mid \text{ es gibt Varianten } C_1, C_2 \text{ von Klauseln aus } M, \\ & \text{so daß } B \text{ eine Resolvente von } C_1, C_2 \text{ ist.}\} \end{aligned}$$

2. $R^0(M) = M$
3. $R^{n+1}(M) = Res(R^n) \cup R^n$
4. M ist unerfüllbar genau dann, wenn es ein n gibt mit $\Box \in R^n(M)$
5. $M \vdash_{\mathbf{R}} A$ gilt genau dann, wenn es ein n gibt mit $\Box \in R^n(M \cup \{\neg A\})$

Beispiel:

Wir wollen die folgende logische Folgerung beweisen:

$$\forall x \forall y (x \subseteq y \leftrightarrow \forall u (u \epsilon x \rightarrow u \epsilon y))$$
$$\models$$
$$\forall x \forall y \forall z (x \subseteq y \wedge y \subseteq z \rightarrow x \subseteq z)$$

Es handelt sich dabei um eine Aussage aus der elementaren Mengenlehre zur Transitivität der Teilmengenbeziehung.

Bemerkenswert ist vielleicht, daß die Transitivität allein aus der Definition der Teilmengenrelation gefolgert werden soll, ohne zusätzliche mengentheoretische Axiome über die ϵ–Relation.

Transformation in Klauselnormalform
- für die Prämisse:

Die Prämisse zerfällt in die beiden Teile

$\forall x \forall y (x \subseteq y \rightarrow \forall u (u \epsilon x \rightarrow u \epsilon y))$

$\forall x \forall y (\forall u (u \epsilon x \rightarrow u \epsilon y) \rightarrow x \subseteq y)$

Die erste Formel wird zu

$$\{\neg conteq(x, y), \neg memb(u, x), memb(u, y)\}$$

mit *conteq* und *memb* für die Infixzeichen $\subseteq$ und ϵ.
Die 2. Formel wird nach Elimination von $\rightarrow$ und Skolemisierung zu

$$\forall x \forall y \exists u ((u \epsilon x \wedge \neg u \epsilon y) \vee x \subseteq y)$$
$$\forall x \forall y ((f(x, y) \epsilon x \wedge \neg f(x, y) \epsilon y) \vee x \subseteq y)$$

Nach Anwendung des Distributivgesetzes:

$$\{memb(f(x, y), x), conteq(x, y)\}, \{\neg memb(f(x, y), y), conteq(x, y)\}$$

- für die Behauptung:

Die Negation der Behauptung führt zu

$$\exists x \exists y \exists z (x \subseteq y \wedge y \subseteq z \wedge \neg x \subseteq z)$$

und nach Einführung von Skolemkonstanten zu den drei Einerklauseln:

$conteq(a, b)$

$conteq(b, c)$

$\neg conteq(a, c)$

Der Resolutionsbeweis:

(1)	$\neg conteq(x, y), \neg memb(u, x), memb(u, y)$	[Vor.]
(2)	$memb(f(x, y), x), conteq(x, y)$	[Vor.]
(3)	$\neg memb(f(x, y), y), conteq(x, y)$	[Vor.]
(4)	$conteq(a, b)$	[¬ Beh.]
(5)	$conteq(b, c)$	[¬ Beh.]
(6)	$\neg conteq(a, c)$	[¬ Beh.]
(7)	$\neg memb(u, a), memb(u, b)$	[4,1]
(8)	$\neg memb(u, b), memb(u, c)$	[5,1]
(9)	$\neg memb(f(a, c), c)$	[6,3]
(10)	$memb(f(a, c), a)$	[6,2]
(13)	$memb(f(a, c), b)$	[7,10]
(19)	$memb(f(a, c), c)$	[8,13]
(20)	$\square$	[19,9]

Dieser Beweis wurde von dem automatischen Beweiser OTTER gefunden.

Der Resolutionskalkül ist auch die Grundlage des Beweisens in Prolog.

Tableaukalkül

- Erweiterung der Formeln auf Vorzeichenformeln wie beim aussagenlogischen Tableaukalkül

- jetzt 4 statt 2 Grundtypen von Vorzeichenformeln:

Typ α:

F	F_1	F_2
$1\neg A$	$0A$	–
$0\neg A$	$1A$	–
$1A \wedge B$	$1A$	$1B$
$0A \vee B$	$0A$	$0B$
$0A \rightarrow B$	$1A$	$0B$

Typ β

F	F_1	F_2
$0A \wedge B$	$0A$	$0B$
$1A \vee B$	$1A$	$1B$
$1A \rightarrow B$	$0A$	$1B$

Typ γ:

F	F_1
$1\forall x A(x)$	$1A(x)$
$0\exists x A(x)$	$0A(x)$

Typ δ:

F	F_1
$1\exists x A(x)$	$1A(x)$
$0\forall x A(x)$	$0A(x)$

Tableauregeln:

α-Regel	$\dfrac{F}{\begin{array}{c}F_1\\F_2\end{array}}$	für α-Formeln F
β-Regel	$\dfrac{F}{F_1 \mid F_2}$	für β-Formeln F
γ-Regel	$\dfrac{F}{F_1(y)}$	für γ-Formeln F und eine neue Variable y
δ-Regel	$\dfrac{F}{F_1(f(x_1,\ldots,x_n))}$	für δ-Formeln F, wobei x_1, ..., x_n alle freien Variablen in F sind und f ein neues n-stelliges Funktionssymbol

Anfangsregel	$\dfrac{}{0A}$	für die zu beweisende Formel A A ohne freie Variable
V-Regel	$\dfrac{}{1B}$	für jedes $B \in M$, B ohne freie Variablen

Def. der Tableaus als Graphen wie im aussagenlogischen Fall.

"geschlossenes Tableau":

Sei T ein Tableau, π ein Pfad in T und σ eine Substitution.

Definition

σ *schließt* π, wenn es

- Formeln B, C gibt, so daß $\sigma(B) = \sigma(C)$, σ kollisionsfrei für B und C ist und $1B, 0C$ auf π liegen oder
- eine der Formeln $0\mathbf{1}$ oder $1\mathbf{0}$ liegt auf π.

σ *schließt* ein Tableau T , wenn σ alle seine Pfade schließt .

Die Abschlußregel oder C-Regel:

Aus einem Tableau T erzeuge ein Tableau T_1
durch Wahl eines Pfades π und einer Substitution σ, die π schließt, und
Anwendung von σ auf das ganze Tableau T.

Beispiel:

$$\begin{array}{cccc}
0\ \forall x\, p(x) \rightarrow \exists y\, p(y) & \text{(Start)} & & 0\ \forall x\, p(x) \rightarrow \exists y\, p(y) \\
| & & & | \\
1\ \forall x\, p(x) & (\alpha\text{-Regel}) & & 1\ \forall x\, p(x) \\
| & & \text{C-Regel} & | \\
0\ \exists y\, p(y) & (\alpha\text{-Regel}) & \Longrightarrow & 0\ \exists y\, p(y) \\
| & & \{Y/X\} & | \\
1\ p(X) & (\gamma\text{-Regel}) & & 1\ p(X) \\
| & & & | \\
0\ p(Y) & (\gamma\text{-Regel}) & & 0\ p(X)
\end{array}$$

Aus der zu beweisenden Aussage entsteht durch Anwendung der α- und γ-Regel das linke Tableau, daraus dann das rechts stehende durch Anwendung der C-Regel.

Korrektheitssatz des Tableaukalküls:

Theorem

Sei $A \in For_\Sigma$, $M \subseteq For_\Sigma$, alle ohne freie Variablen
Wenn es ein geschlossenes Tableau für A über M gibt, dann ist $M \models A$.

Beweis: s. Schmitt (2008), S. 168.

Vollständigkeitssatz des Tableaukalküls:

Theorem

Sei A eine Formel und M eine Menge von Formeln jeweils ohne freie Variablen.

Gilt $M \models A$

dann terminiert jedes

- *faire Verfahren,*
- *das mit* $0A$ *und* $\sigma_0 = id$ *beginnt,*
- *und die Konstruktionsvorschrift einhält*

nach endlich vielen Schritten in einem geschlossenen Tableau.

Fairness bedeutet, daß auf jedem Pfad, jede mögliche Regelanwendung auch schließlich stattfindet.
Insbesondere wird auf jedem offenen Pfad jede γ-Formel unbeschränkt oft benutzt.

Der *Beweis* (s. Schmitt 2008, S. 175f.) verwendet das Lemma von König: In jedem unendlichen Baum mit endlichen Knotengraden existiert ein unendlicher Pfad.

Sequenzenkalkül:

kann ebenfalls auf die Prädikatenlogik übertragen werden.

Modale Logik

Motivation:

Im Unterschied zur klassischen Logik,
in der nur die Wahrheit einer Aussage von Bedeutung ist,
spielt in der modalen Logik
die Art und Weise, der *Modus*,
in der eine Aussage wahr ist eine große Rolle.

Eine Aussage ist

- notwendigerweise wahr, zufälligerweise wahr
- heute, gestern oder morgen wahr
- wird geglaubt, gehört zum Wissen einer Person
- ist vor/nach einer Aktion wahr, nach Ausführung eines Programms wahr.

Formalisierung:

Definition

1. $\mathbf{1}, \mathbf{0} \in mFor0_\Sigma$
2. Jede aussagenlogische Variable $P \in \Sigma$ ist in $mFor0_\Sigma$.
3. Mit $A, B \in mFor0_\Sigma$ liegen ebenfalls in $mFor0_\Sigma$:
   $\neg A$, $A \wedge B$, $A \vee B$, $A \rightarrow B$.
4. Mit $A \in mFor0_\Sigma$ liegen ebenfalls in $mFor0_\Sigma$:
   $\Box A$ (gelesen als „Box A“, „notwendig A“)
   $\Diamond B$ (gelesen als „Diamond A“, „möglich A“)

Die Interpretation ist jetzt nicht mehr einfach eine boolesche Variablenbelegung, sondern hängt vom Zustand aus einer Zustandsmenge ab. Genauer:

Definition

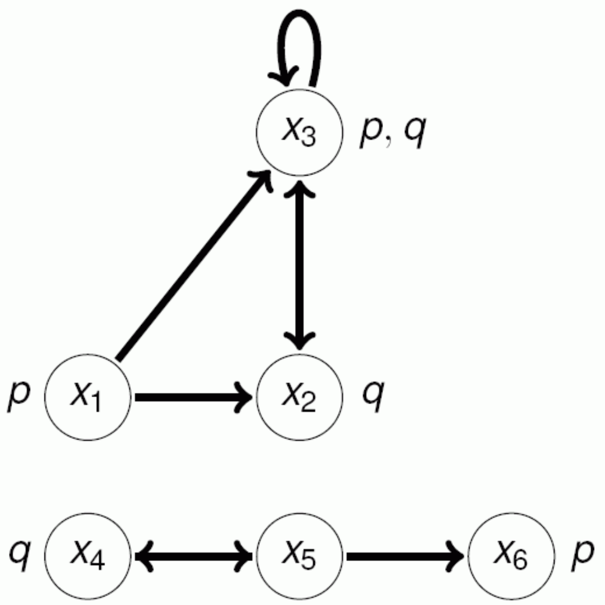
Sei Σ eine Menge aussagenlogischer Variablen.
Eine Kripke-Struktur

$$\mathcal{K} = (S, R, I)$$

über Σ besteht aus:

- S eine nichtleere Menge
  (die Menge von *Zuständen* oder möglichen *Welten*)
- R $\subseteq S \times S$ (die Zugänglichkeitsrelation)
- I: $(\Sigma \times S) \to \{W, F\}$ (Interpretation der AL-Variablen)

Beispiel einer Kripke-Struktur:



Menge der Zustände $S = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ $R = \{(x_1, x_2), (x_1, x_3), (x_2, x_3), (x_3, x_2), (x_3, x_3), (x_4, x_5), (x_5, x_4), (x_5, x_6)\}$
I(p, x_1) = I(p, x_3) = I(p, x_6) = 1, sonst 0;
I(q, x_2) = I(q, x_3) = I(q, x_4) = 1, sonst 0.

Auswertung von Formeln:

Sei $\mathcal{K} = (S, R, I)$ eine Kripke-Struktur. Wir definieren für jeden Zustand $s \in S$, wann eine Formeln aus $mFor0$ in s wahr ist.

Definition

$$val_s(\Box A) = \begin{cases} W & \text{falls für alle } s' \in S \text{ mit } sRs' \\ & \text{gilt } val_{s'}(A) = W \\ F & \text{sonst} \end{cases}$$

$$val_s(\Diamond A) = \begin{cases} W & \text{falls ein } s' \in S \text{ existiert mit } sRs' \\ & \text{und } val_{s'}(A) = W \\ F & \text{sonst} \end{cases}$$

Notation

$\mathcal{K} = (S, R, I)$ eine Kripke-Struktur,
$s \in S$,
F eine modale Formel

$$(\mathcal{K}, s) \models F \Leftrightarrow val_s(F) = W$$

wenn $\mathcal{K}$ aus dem Kontext bekannt ist auch:

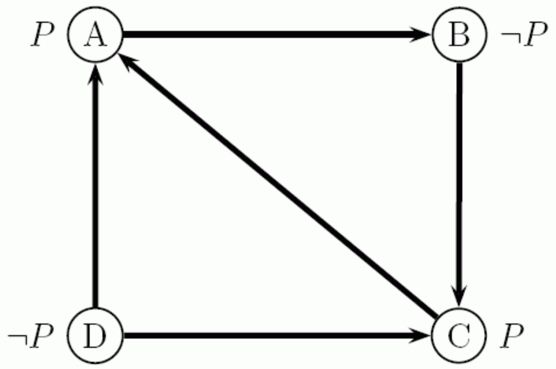
$$s \models F \Leftrightarrow val_s(F) = W$$

$$\mathcal{K} \models F \Leftrightarrow \text{für alle } s \in S \text{ gilt } (\mathcal{K}, s) \models F$$

Gültigkeit in einen Kripke-Rahmen (S, R) :

$$(S, R) \models F \Leftrightarrow \text{für alle } I \text{ gilt } (S, R, I) \models F$$

Beispiel zur Auswertung von Formeln:



$(\mathcal{K}, A) \models P$	$(\mathcal{K}, B) \models P$	$(\mathcal{K}, C) \models P$	$(\mathcal{K}, D) \models P$
$(\mathcal{K}, A) \models \Box P$	$(\mathcal{K}, B) \models \Box P$	$(\mathcal{K}, C) \models \Box P$	$(\mathcal{K}, D) \models \Box P$
$(\mathcal{K}, A) \models \Box\Box P$	$(\mathcal{K}, B) \models \Box\Box P$	$(\mathcal{K}, C) \models \Box\Box P$	$(\mathcal{K}, D) \models \Box\Box P$

true false

Logisches Folgern in der Modallogik

Definition

Sei A eine Formel und Γ eine Menge von Formeln der modalen Aussagenlogik.

A ist eine logische Folgerung aus Γ

$$\Gamma \vdash A$$

gdw

für alle Kripke-Strukturen $\mathcal{K}$ und jede Welt s von $\mathcal{K}$ gilt

wenn $(\mathcal{K}, s) \models \Gamma$ dann auch $(\mathcal{K}, s) \models A$

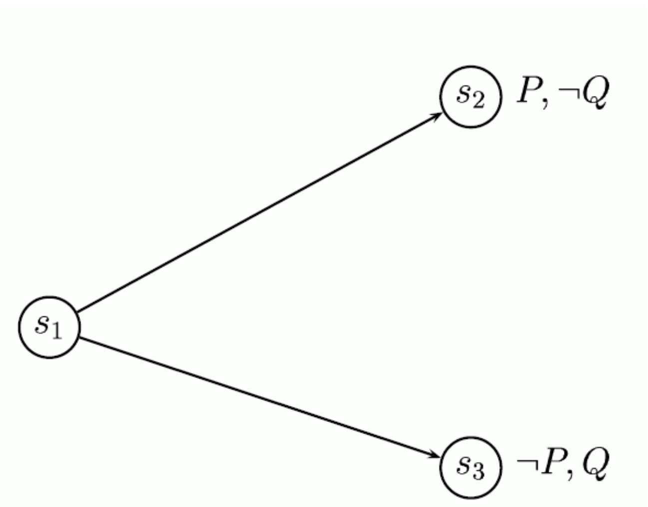
A ist allgemeingültig wenn

$$\emptyset \vdash A$$

allgemeingültige Formeln

1. $\Box(P \to Q) \to (\Box P \to \Box Q)$
2. $(\Box P \wedge \Box(P \to Q)) \to \Box Q$
3. $(\Box P \vee \Box Q) \to \Box(P \vee Q)$
4. $(\Box P \wedge \Box Q) \leftrightarrow \Box(P \wedge Q)$
5. $\Box P \leftrightarrow \neg\Diamond\neg P$
6. $\Diamond(P \vee Q) \leftrightarrow (\Diamond P \vee \Diamond Q)$
7. $\Diamond(P \wedge Q) \to (\Diamond P \wedge \Diamond Q)$

Gegenbeispiel zur Allgemeingültigkeit von $\Box(P \vee Q) \to (\Box P \vee \Box Q)$:



Relative Allgemeingültigkeit

Die Formel

$$\Box A \rightarrow A$$

ist nicht allgemeingültig.

Aber

für alle Kripke-Strukturen $\mathcal{K} = (S, R, I)$, so daß (S, R) eine reflexive Relation ist gilt

$$\mathcal{K} \models \Box A \rightarrow A$$

allgemeingültige Formel	Eigenschaft von R
$\Box A \rightarrow A$	reflexiv
$\Box A \rightarrow \Box\Box A$	transitiv
$A \rightarrow \Box\Diamond A$	symmetrisch
$\Box\Box A \rightarrow \Box A$	dicht

("R dicht" heißt: für alle Zustände x, z mit $R(x, z)$ existiert ein Zustand y mit $R(x, y)$ und $R(y, z)$.)

allgemeingültige Formel	Eigenschaft von R
$\Box p \rightarrow p$	reflexiv
$p \rightarrow \Diamond p$	reflexiv
$\Box\Box p \rightarrow \Box p$	reflexiv
$\Box\Diamond p \rightarrow \Diamond p$	reflexiv
$\Box p \rightarrow \Diamond\Box p$	reflexiv
$\Diamond\Diamond p \rightarrow \Diamond p$	transitiv
$\Box p \rightarrow \Box\Box p$	transitiv
$p \rightarrow \Box\Diamond p$	symmetrisch
$\Box\Box p \leftrightarrow \Box p$	reflexiv und transitiv
$\Diamond\Diamond p \leftrightarrow \Diamond p$	reflexiv und transitiv
$\Diamond\Box p \leftrightarrow \Box p$	Äquivalenzrelation
$\Box\Diamond p \leftrightarrow \Diamond p$	Äquivalenzrelation

Umkehrung dieser Art von Beziehungen:

Gilt für einen Kripke-Rahmen (S, R)

$$\text{für alle } I \text{ gilt } (S, R, I) \models \Box A \to A$$

dann ist

$$(S, R) \text{ reflexiv}$$

Charakterisierungstheorie

Definition

Sei $\mathbf{R}$ eine Klasse von Kripke-Rahmen,
und F eine Formel der Modallogik.

F charakterisiert die Klasse $\mathbf{R}$ genau dann, wenn für alle Kripke-Rahmen (S, R) gilt

$$\text{für alle } I \text{ gilt } (S, R, I) \models F$$
$$gdw$$
$$(S, R) \in \mathbf{R}$$

Einige Charakterisierungsresultate:

Formel	charakterisierte Eigenschaft
$\Box A \to A$	reflexiv
$\Box A \to \Box\Box A$	transitiv
$A \to \Box\Diamond A$	symmetrisch
$\Box\Box A \to \Box A$	dicht
$\Diamond A \to \Box A$	partiell funktional
$\Box A \to \Diamond A$	endlos

Grenzen der Charakterisierungstheorie:

Konkretisierung

Sei ϕ eine Formel der Prädikatenlogik in der Signatur $\Sigma = \{R\}$ und

$$\mathcal{R}_\phi = \{(S, R) \mid (S, R) \models \phi\}$$

Frage 1 Gibt es zu jedem ϕ eine modallogische Formel F, so daß die Klasse der Rahmen $\mathcal{R}_\phi$ charakterisiert?

Frage 2 Gibt es zu jeder modallogischen Formel F eine prädikatenlogische Formel ϕ, so daß $\mathcal{R}_\phi$ mit der Klasse der durch F charakterisierten Rahmen zusammenfällt?

Antworten:

Antwort 1 Nein

Z.B. für $\phi = \forall x \neg R(x, x)$ kann die Klasse $\mathcal{R}_\phi$ nicht durch eine modallogische Formel charakterisiert werden

Antwort 2 Nein

Es gibt modallogische Formeln F, so dass die durch F charakterisierten Rahmen nicht durch eine prädikatenlogische Formel axiomatisiert werden können.

Entscheidbarkeit der Modallogik

Theorem

Jede Menge Γ modallogischer Formeln, die überhaupt ein Modell hat,
hat auch ein Modell (S, R, I), so dass S endlich ist, wobei eine obere Schranke für die Größe von S aus Γ berechnet werden kann.

Korollar

Die modale Aussagenlogik **K** ist entscheidbar, d.h.

es gibt einen Algorithmus, der für jede Formel A entscheidet, ob A eine **K**-Tautologie ist oder nicht.

Informale Interpretationen der modallogischen Operatoren:

$\Box F$
F ist notwendigerweise wahr
F ist zu jedem zukünftigen Zeitpunkt wahr
Ein Agent a glaubt F
Ein Agent a weiß F
Nach jeder Ausführung des Programms p gilt F

Falls erforderlich schreibt man

$$\Box_a F, \quad \Box_p F, \quad [a]F \quad \text{oder} \quad [p]F$$

anstelle von $\Box F$.

$\Diamond F \equiv \neg\Box\neg F$	
F ist notwendigerweise wahr	F ist möglicherweise wahr
F ist zu jedem zukünftigen Zeitpunkt wahr	es gibt einen zukünftigen Zeitpunkt, zu dem F wahr ist.
Ein Agent a glaubt F	F ist konsistent mit den Aussagen, die a für wahr hält.
Ein Agent a weiß F	a weiß nicht, daß F falsch ist.
Nach jeder Ausführung des Programms p gilt F	Es gibt eine Ausführung des Programms p, nach der F wahr ist.

Je nach Interpretation gelten unterschiedliche Allgemeingültigkeiten:

$\Box F$	$\Box F \rightarrow F$	$\Box F \rightarrow \Box\Box F$	$\Box F \rightarrow \Diamond F$	$(\Box(F \rightarrow G) \wedge \Box F) \rightarrow \Box G$	$\Diamond true$
F ist notwendigerweise wahr	yes	yes	yes	yes	yes
F ist immer wahr	no	yes	no	yes	no
Ein Agent a glaubt F	no	yes	yes	yes	yes
Ein Agent a weiß F	yes	yes	yes	yes	yes
Nach jeder Ausführung des Programms p gilt F	no	no	no	yes	no

Lineare temporale Logik

Spezialfall der Modallogik mit weiteren Operatoren

zunächst Formalisierung des Raumes der möglichen Zeitpunkte:
"Omega-Struktur"

Definition

Eine omega-Struktur $\mathcal{R} = (\mathbb{N}, <, \xi)$ für eine aussagenlogische Signatur P besteht aus der geordneten Menge der natürlichen Zahlen

$$(\mathbb{N}, <)$$

interpretiert als Menge abstrakter Zeitpunkte und einer Funktion

$$\xi : \mathbb{N} \to 2^P$$

mit der Intention

$$p \in \xi(n) \Leftrightarrow \text{ in } \mathcal{R} \text{ ist } p \text{ zum Zeitpunkt } n \text{ wahr}$$

ξ_n steht für das bei n beginnende Endstück von ξ:

$$\xi_n(m) = \xi(n+m) \qquad \text{insbes. } \xi_0 = \xi$$

Formeln der linearen temporalen Logik (LTL):

wie modallogische Formeln, es kommen die Operatoren **U** ("until") (binär) und X (unär) hinzu.

LTL-Formeln

Definition

Σ eine Menge aller AL-Atome. *LTLFor* wird definiert durch

1. $\Sigma \subseteq LTLFor$
2. $\mathbf{1}, \mathbf{0} \in LTLFor$
3. Liegen A, B in *LTLFor*, dann auch alle aussagenlogischen Kombinationen von A und B.
4. für $A, B \in LTLFor$ gilt auch
   1. $\Box A \in LTLFor$ und
   2. $\Diamond B \in LTLFor$ und
   3. $A\ \mathbf{U}\ B \in LTLFor$
   4. $X\ A$

Die Symbole $\Box$, $\Diamond$, X und $\mathbf{U}$ heißen temporale Modaloperatoren.

Semantik der LTL:

Definition

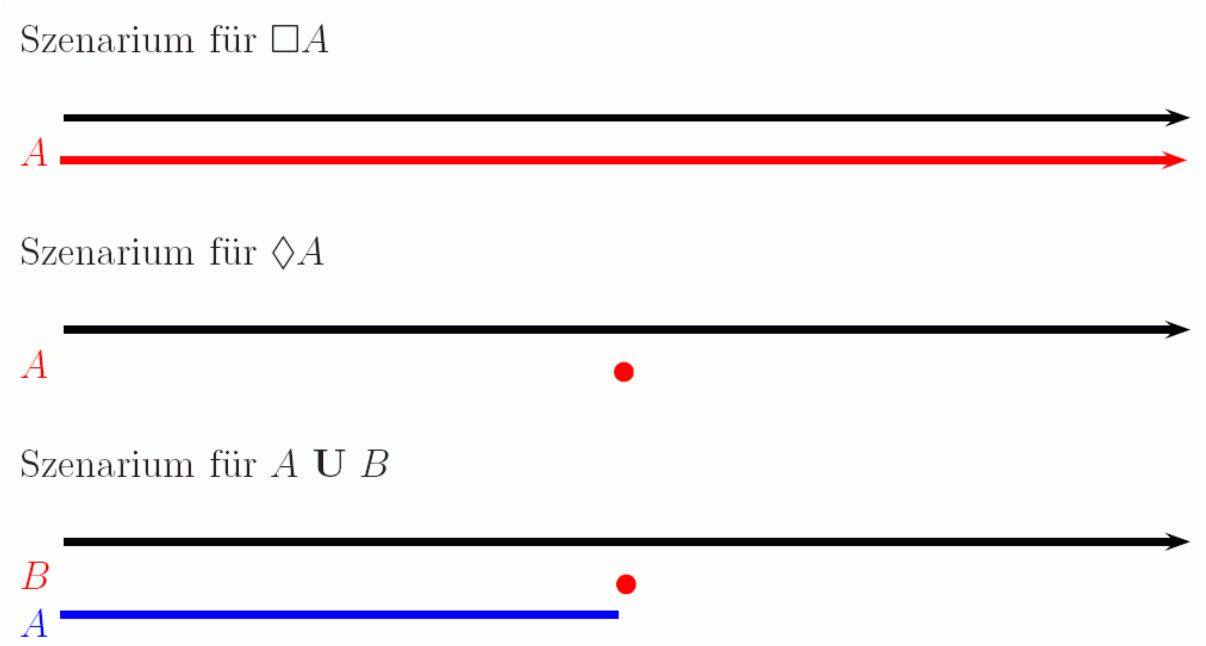
Sei $\mathcal{R} = (\mathbb{N}, <, \xi)$ eine omega-Struktur und A eine *LTL* Formel.

$\xi \models p$	gdw	$p \in \xi(0)$ (p ein AL Atom)
$\xi \models op(A, B)$		für AL-Kombinationen $op(A, B)$ von A und B wie üblich
$\xi \models \Box A$	gdw	für alle $n \in \mathbb{N}$ gilt $\xi_n \models A$
$\xi \models \Diamond A$	gdw	es gibt ein $n \in \mathbb{N}$ mit $\xi_n \models A$
$\xi \models A\ \mathbf{U}\ B$	gdw	es gibt $n \in \mathbb{N}$ mit $\xi_n \models B$ und für alle m mit $0 \leq m < n$ gilt $\xi_m \models A$
$\xi \models X\ A$	gdw	$\xi_1 \models A$

Reduktion auf **U** und **1** ist möglich:

$$\Diamond A \leftrightarrow \mathbf{1}\ \mathbf{U}\ A$$
$$\Box A \leftrightarrow \neg(\mathbf{1}\ \mathbf{U}\ \neg A)$$

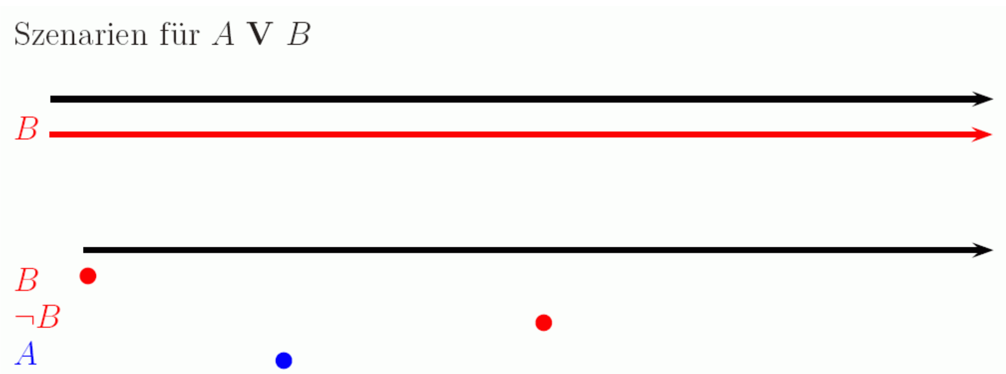
Visualisierung der LTL-Semantik mit Zeitstrahl:



Zusätzliche Operatoren:

$\xi \models A\ \mathbf{U}_w\ B$	gdw	für alle $n \in \mathbb{N}$ gilt $\xi_n \models (A \wedge \neg B)$ oder es gibt $n \in \mathbb{N}$ mit $\xi_n \models B$ und für alle m mit $0 \leq m < n$ gilt $\xi_m \models A$
$\xi \models A\ \mathbf{V}\ B$	gdw	$\xi \models B$ und für alle $n \in \mathbb{N}$ gilt falls $\xi_n \models \neg B$ dann gibt es ein m mit $0 \leq m < n$ und $\xi_m \models A$

Visualisierung von **V**:



allgemeingültige Aussagen in LTL:

1. $A \mathbf{U} B \leftrightarrow (A \mathbf{U}_w B) \wedge \Diamond B$
2. $A \mathbf{U}_w B \leftrightarrow A \mathbf{U} B \vee \Box(A \wedge \neg B)$
3. $A \mathbf{V} B \leftrightarrow \neg(\neg A \mathbf{U} \neg B)$
4. $A \mathbf{U} B \leftrightarrow (B \vee (A \wedge X (A \mathbf{U} B)))$
5. $A \mathbf{V} B \leftrightarrow (B \wedge A) \vee (B \wedge X (A \mathbf{V} B))$

Ein Beispiel:

Sei p ein aussagenlogisches Atom.
Gesucht ist eine LTL-Formel A_{2p}, so daß für jedes ξ gilt

$$\xi \models A_{2p} \quad \text{gdw} \quad (n \text{ ist gerade} \Leftrightarrow p \in \xi(n))$$

$$A_{2p} = p \wedge X \neg p \wedge \Box(p \leftrightarrow XX\, p)$$

Erstaunlicherweise gibt es keine LTL-Formel A mit

$$\xi \models A \quad \text{gdw} \quad (n \text{ ist gerade} \Rightarrow p \in \xi(n))$$

Anwendungsbeispiel:

```
REQUIREMENT: When a connection is made to the SMTP
   server, all queued messages in the OutBox mail
   will be transferred to the server.
REFINEMENT: Before QueuedMailSent, SMTPServerConnected
PATTERN: Existence
SCOPE: Before
AlTERNATE: Global Precedence
PARAMETERS: Propositional
LTL:  <>QueuedMailSent ->
           (!QueuedMailSent U SMTPServerConnected)
SOURCE: Jeff Isom \cite{isom:98}
DOMAIN: GUI
```

Ausschnitte entnommen aus
Beckert (2010) und Schmitt (2008)
(genaue Quellenangaben siehe **http://www.uni-forst.gwdg.de/~wkurth/fs10_lit.htm**)