Vorlesung Logiksysteme

Martin Mundhenk

Univ. Jena, Institut für Informatik
13. Februar 2018

Wintersemester 2017/18

Inhalt dieser Vorlesung

- Was wird betrachtet?

  1. Aussagenlogik
  2. Nicht-klassische Aussagenlogiken
     - Modallogik
     - Temporale Logik

- Wie wird es betrachtet?

  Schwerpunkt auf Beweissystemen und deren Vollständigkeit
  - Wie lassen sich gültige Formeln formal beschreiben?
  - Wie gewinnt man Algorithmen aus formalen Beschreibungen?

- Ziel: Verinnerlichung der Idee, was ein Beweis ist;
  Verständnis der Vollständigkeitsbeweise
  und Befähigung zum selbständigen Führen „kleiner“ Beweise

Vorlesung Logiksysteme (Winter 2017/18)

Teil 1: Aussagenlogik

1. Aussagenlogik

VL01: Umgangssprachliche und formale Aussagenlogik
VL02: Äquivalente Formeln und adäquate Verknüpfungszeichen

VL03: Ein Tableau-Kalkül
VL04: Ein Frege-Kalkül
VL05: Die Vollständigkeitssätze für die Kalküle

[Literatur (siehe Semesterapparat):

Schöning: Logik für Informatiker	VL01/02
Priest: An Introduction to Non-classical Logic	VL01/02/03/05
Nerode, Shore: Logic for Applications	VL03/05
Mendelson: Introduction to Mathematical Logic	VL04/05]

1.1+2 Grundbegriffe der Aussagenlogik

Zuerst werden wir uns aussagenlogische Konstrukte in der Umgangssprache anschauen.

Dann werden die Grundbegriffe der (formalen) Aussagenlogik definiert:

- Wie sehen aussagenlogische Formeln aus?
- Was ist eine Belegung und wann erfüllt sie eine Formel?
- Was sind gültige, erfüllbare und unerfüllbare Formeln?
- Wann sind Formeln äquivalent?
- Welche Verknüpfungszeichen braucht man überhaupt in Formeln? (adäquate Verknüpfungszeichen)

Vorlesung 1:

Umgangssprachliche und formale Aussagenlogik

Wir benutzen Logik in der Umgangssprache.
Daraus hat sich die formale Logik entwickelt.

Wir wollen uns klarmachen, wann wir umgangssprachliche Logik und wann wir formale Logik benutzen.

1.1 Umgangssprachliche Aussagenlogik

Einführendes Beispiel

Aussagenlogik betrachtet Aussagen und deren Zusammensetzung, und untersucht das korrekte Schlussfolgern.

Beispiel:

(1) Ingo trifft Peter oder Maria.
(2) Wenn er Peter trifft, dann trifft er Vera oder Andreas.
(3) Wenn er Vera trifft, dann trifft er auch Maria.
(4) Wenn er Vera nicht trifft, dann trifft er nicht Andreas.

Ist „Ingo trifft Maria“ eine korrekte Folgerung aus (1)–(4)?

Aussagen

Eine ugs. Aussage ist ein Satz, der entweder wahr oder falsch ist.

Aus einzelnen Aussagen können neue Aussagen gebildet werden,
indem Aussagen sprachlich so verknüpft werden,
dass neue Aussagen entstehen.
Ob die neue Aussage wahr oder falsch ist,
ergibt sich aus den Wahrheitswerten der Aussagen,
aus denen sie gebildet wird.
Dazu geben wir für jede Verknüpfung in einer Wahrheitswertetabelle an,
wie sich der Wahrheitswert der neuen Aussage
aus den Wahrheitswerten der verknüpften Aussagen ergibt.

(Statt „Aussage α ist wahr" sagt man
z.B. auch „α ist korrekt" oder „α gilt".)

Umgangssprachliche Negation

Die Negation einer ugs. Aussage α
ist z.B. „Nicht α“ oder „α ist falsch“.

Die Negation der Aussage $x = 2$ schreibt man auch $x \neq 2$.
Diese „Negation durch Durchstreichen“ macht man gerne mit formalen Symbolen.
Das werden wir später auch so machen.

Wenn α wahr ist, dann ist „α ist falsch“ falsch,
und wenn α falsch ist, dann ist „α ist falsch“ wahr.

Das schreiben wir als Wahrheitswertetabelle auf:

α	Nicht α
wahr	falsch
falsch	wahr

Umgangssprachliche Konjunktion

Die Konjunktion zweier ugs. Aussagen α und β
ist die Aussage „α und β“.

Zum schnellen oder kurzen Aufschreiben schreibt man „α und β“ auch als „α & β“.

Die Wahrheitswertetabelle für die Konjunktion ist:

α	β	α und β
wahr	wahr	wahr
wahr	falsch	falsch
falsch	wahr	falsch
falsch	falsch	falsch

Umgangssprachliche Disjunktion

Die Disjunktion zweier ugs. Aussagen α und β
ist die Aussage „α oder β“.

Die Wahrheitswertetabelle für die Disjunktion ist:

α	β	α oder β
wahr	wahr	wahr
wahr	falsch	wahr
falsch	wahr	wahr
falsch	falsch	falsch

Man beachte, dass die Bedeutung der Aussage „Entweder α oder β“ anders ist als die von „α oder β“.

Umgangssprachliche Implikation

Die Implikation einer ugs. Aussage β aus einer ugs. Aussage α ist die Aussage „Wenn α, dann β“ oder „Aus α folgt β“.

Zum schnellen oder kurzen Aufschreiben schreibt man „Wenn α, dann β“ auch als „$\alpha \Rightarrow \beta$“.

Die Wahrheitswertetabelle für die Implikation ist:

α	β	wenn α, dann β
wahr	wahr	wahr
wahr	falsch	falsch
falsch	wahr	wahr
falsch	falsch	wahr

Umgangssprachliche Äquivalenz

Die Äquivalenz zweier ugs. Aussagen α und β
ist die Aussage „α genau dann, wenn β“.

Zum schnellen oder kurzen Aufschreiben schreibt man
„α genau dann, wenn β“ auch als „α gdw. β“ oder als „$\alpha \Leftrightarrow \beta$“.

Die Wahrheitswertetabelle für die Äquivalenz ist:

α	β	α genau dann, wenn β
wahr	wahr	wahr
wahr	falsch	falsch
falsch	wahr	falsch
falsch	falsch	wahr

Zurück zum Beispiel

(1) Ingo trifft Peter oder Maria. (P oder M.)
(2) Wenn er Peter trifft, dann trifft er Vera oder Andreas. (Wenn P, dann (V oder A).)
(3) Wenn er Vera trifft, dann trifft er auch Maria. (Wenn V, dann M.)
(4) Wenn er Vera nicht trifft, dann trifft er nicht Andreas. (Wenn nicht V, dann nicht A.)

Ist „Ingo trifft Maria" (M) eine Folgerung aus (1)–(4)?

D.h.: ist die „große" Aussage

„Wenn (P oder M) und
(wenn P, dann (V oder A)) und
(wenn V, dann M) und
(wenn nicht V, dann nicht A),
dann M."

wahr ? Dazu schauen wir uns die Wahrheitswertetabelle der großen Aussage an . . .

P	M	V	A	(1)	(2)	(3)	(4)	große Aussage
wahr	wahr	wahr	wahr	wahr	wahr	wahr	wahr	wahr
wahr	wahr	wahr	falsch	wahr	wahr	wahr	wahr	wahr
wahr	wahr	falsch	wahr	wahr	wahr	wahr	falsch	wahr
wahr	wahr	falsch	falsch	wahr	falsch	wahr	wahr	wahr
wahr	falsch	wahr	wahr	wahr	wahr	falsch	wahr	wahr
wahr	falsch	wahr	falsch	wahr	wahr	falsch	wahr	wahr
wahr	falsch	falsch	wahr	wahr	wahr	wahr	falsch	wahr
wahr	falsch	falsch	falsch	wahr	falsch	wahr	wahr	wahr
falsch	wahr	wahr	wahr	wahr	wahr	wahr	wahr	wahr
falsch	wahr	wahr	falsch	wahr	wahr	wahr	wahr	wahr
falsch	wahr	falsch	wahr	wahr	wahr	wahr	falsch	wahr
falsch	wahr	falsch	falsch	wahr	wahr	wahr	wahr	wahr
falsch	falsch	wahr	wahr	falsch	wahr	falsch	wahr	wahr
falsch	falsch	wahr	falsch	falsch	wahr	falsch	wahr	wahr
falsch	falsch	falsch	wahr	falsch	wahr	wahr	falsch	wahr
falsch	falsch	falsch	falsch	falsch	wahr	wahr	wahr	wahr

Also: die „große Aussage“ („Ingo trifft Maria“ ist eine Folgerung aus (1)–(4)) ist wahr!

1.2 Formale Aussagenlogik

Die formale Aussagenlogik ist ein abstraktes Modell der umgangssprachlichen Aussagenlogik.

Zuerst modelliert man die Zusammensetzung von Aussagen (Syntax). Das ergibt die *Formeln* der Aussagenlogik (wie eine formale Sprache).

Anschließend modelliert man die Wahrheitswerte durch eine *Erfüllungsrelation* (Semantik).

Abschließend werden semantische Eigenschaften von Formeln definiert und betrachtet.

Definition 1.1 (die Sprache der Aussagenlogik: Formeln)

Eine atomare Formel (kurz: Atom) hat die Form A_i für $i = 0, 1, 2, \ldots$

(Aussagenlogische) Formeln sind induktiv definiert wie folgt.

1. Die Konstanten $\top$ (*verum*) und $\bot$ (*falsum*) und alle Atome sind Formeln.

2. Für alle Formeln α ist $\neg\alpha$ (*Negation von* α) ebenfalls eine Formel.
Für alle Formeln α und β sind

$(\alpha \wedge \beta)$ (*Konjunktion von* α *und* β*, logisches Und*),
$(\alpha \vee \beta)$ (*Disjunktion von* α *und* β*, logisches Oder*) und
$(\alpha \to \beta)$ (*Implikation von* α *und* β*, logisches Wenn … dann*)

ebenfalls Formeln.

(3. Es gibt keine anderen Formeln.)

Die Begriffe *Aussage*, *Wahrheitswert*, *wahr* und *falsch* werden in der formalen Logik nicht verwendet!

Wir haben bereits gesehen, wie die Form von ugs. Aussagen durch Formeln modelliert wird.

Nun brauchen wir noch eine Modellierung der Wahrheitswerte.
Dazu definieren wir die *Belegung* und die *Erfüllungsrelation*.

Definition 1.2 (Belegung)

Eine Belegung $\mathcal{B}$ ist eine Menge $\mathcal{B} \subseteq \{A_0, A_1, A_2, \ldots\}$ von Atomen.

Der Wahrheitswert der Aussage „$A_i \in \mathcal{B}$" modelliert,
dass A_i für eine wahre Aussage steht – bezogen auf die Welt $\mathcal{B}$.

Die Erfüllungsrelation modelliert,
wie sich Wahrheitswerte in zusammengesetzten Aussagen übertragen

...

Definition 1.3 (Erfüllungsrelation $\models$)

Sei $\mathcal{B}$ eine Belegung, α und β seien Formeln.

Die Relation $\models$ zwischen Belegungen und Formeln ist wie folgt definiert.

$$
\begin{array}{rcll}
\mathcal{B} & \models & \top & \\
\mathcal{B} & \not\models & \bot & \\
\mathcal{B} & \models & A_i & \text{gdw.} \quad A_i \in \mathcal{B}, \text{ für atomare Formeln } A_i \\
\mathcal{B} & \models & \neg\alpha & \text{gdw.} \quad \mathcal{B} \not\models \alpha \\
\mathcal{B} & \models & (\alpha \wedge \beta) & \text{gdw.} \quad \mathcal{B} \models \alpha \text{ und } \mathcal{B} \models \beta \\
\mathcal{B} & \models & (\alpha \vee \beta) & \text{gdw.} \quad \mathcal{B} \models \alpha \text{ oder } \mathcal{B} \models \beta \\
\mathcal{B} & \models & (\alpha \rightarrow \beta) & \text{gdw.} \quad \mathcal{B} \not\models \alpha \text{ oder } \mathcal{B} \models \beta
\end{array}
$$

Man spricht die Aussage $\mathcal{B} \models \varphi$ als „$\mathcal{B}$ erfüllt φ“ aus.

Für „Nicht $\mathcal{B} \models \varphi$“ schreibt man $\mathcal{B} \not\models \varphi$.

Eigenschaften von Formeln

Von besonderem Interesse sind Formeln,
die von jeder Belegung erfüllt werden.

Definition 1.4 (gültig, erfüllbar)

1. Eine Formel α heißt gültig (oder Tautologie), wenn α von jeder Belegung erfüllt wird (Schreibweise: $\models \alpha$).
2. Eine Formel heißt erfüllbar, wenn es eine Belegung gibt, die sie erfüllt.
   Anderenfalls heißt die Formel unerfüllbar (oder Kontradiktion).

Semantische Folgerung

Eine Verallgemeinerung von $\models$

Für Formelmengen Γ bedeutet $\mathcal{B} \models \Gamma$ („$\mathcal{B}$ erfüllt Γ"),
dass $\mathcal{B} \models \varphi$ für alle $\varphi \in \Gamma$ gilt.

Definition 1.5 (Semantische Folgerung)

Sei Γ eine Formelmenge und φ eine Formel.
Die Relation $\Vdash\!\models$ zwischen Formelmengen und Formeln ist wie folgt definiert:

$\Gamma \Vdash\!\models \varphi$ genau dann, wenn
jede Belegung, die Γ erfüllt, ebenfalls φ erfüllt.

(D.h.: $\Gamma \Vdash\!\models \varphi$ gdw. für jede Belegung $\mathcal{B}$ gilt: wenn $\mathcal{B} \models \Gamma$, dann $\mathcal{B} \models \varphi$.)

Die Aussage $\Gamma \Vdash\!\models \varphi$ spricht man
„φ ist Folgerung von Γ" oder „aus Γ folgt φ" aus.

Schreibweisen:

- Mengenklammern und Vereinigungszeichen lässt man gerne weg: z.B. schreibt man $\alpha_1, \ldots, \alpha_n \Vdash\!\models \varphi$ oder $\Gamma, \alpha \Vdash\!\models \varphi$.
- Statt $\emptyset \Vdash\!\models \varphi$ schreibt man $\Vdash\!\models \varphi$.

Lemma 1.6 ($\Vdash\!\models$ verallgemeinert $\models$)

Sei φ eine Formel. Dann gilt: $\Vdash\!\models \varphi$ genau dann, wenn $\models \varphi$.

Lemma 1.7 (Zusammenhang zwischen $\Vdash$ und $\to$)

Sei $n \in \mathbb{N}^+$, und $\alpha_1, \ldots, \alpha_n, \varphi$ seien Formeln.
Die folgenden Aussagen sind äquivalent für alle $i = 1, 2, \ldots, n+1$.

1. $\alpha_1, \ldots, \alpha_n \quad \Vdash \varphi$
2. $\alpha_1, \ldots, \alpha_{i-1} \Vdash (\alpha_i \to (\alpha_{i+1} \to \ldots (\alpha_n \to \varphi) \ldots))$
3. $\alpha_1, \ldots, \alpha_{i-1} \Vdash ((\bigwedge_{j=i}^{n} \alpha_j) \to \varphi)$

Insbesondere gelten also folgende Äquivalenzen zu 1.–3. ($i = 1$) :

- $\models (\alpha_1 \to (\alpha_2 \to \ldots (\alpha_n \to \varphi) \ldots))$
- $\models ((\bigwedge_{i=1}^{n} \alpha_i) \to \varphi)$

Außerdem folgt: $\quad \alpha \Vdash \varphi$ gdw. $\models \alpha \to \varphi$.

Was haben wir in Vorlesung 1 gelernt?

- Wir haben die *umgangssprachliche* Aussagenlogik aus Aussagen, Aussageverknüpfungen „nicht“, „und“, „oder“, „wenn ... dann...“ und „... genau dann, wenn ...“ kennengelernt und gesehen, wie sich die Wahrheitswerte „wahr“ und „falsch“ von Aussagen auf Aussageverknüpfungen übertragen.
  Das haben wir mit Hilfe von Wahrheitswertetabellen aufgeschrieben.
- Wir haben die *formale* Aussagenlogik kennengelernt.
  Wir kennen Formeln, die induktiv aus Atomen, $\bot$, $\top$ und den Verknüpfungszeichen $\neg$, $\wedge$, $\vee$ und $\rightarrow$ aufgebaut sind.
  Wir wissen, was eine Belegung ist,
  und kennen die Erfüllungsrelation $\models$ zwischen Belegungen und Formeln, die induktiv über den Aufbau der Formeln definiert ist.
- Wichtig: wir wollen die Begriffe der umgangssprachlichen Logik mit denen der formalen Logik nicht durcheinanderbringen.
- Wir können Formeln die Eigenschaften erfüllbar, gültig und unerfüllbar zuordnen.
- Wir kennen die semantische Folgerung $\Vdash$ als Verallgemeinerung von $\models$.

Vorlesung 2: Äquivalente Formeln und adäquate Verknüpfungszeichen

Unterschiedliche Formeln können von genau den gleichen Belegungen erfüllt werden – sie sind also semantisch gleich. Beim „Rechnen" mit Formeln kann man sie gegenseitig austauschen und ggf. das Rechnen vereinfachen.

Semantisch gleiche Formeln heißen *äquivalent* – diesen Begriff werden wir zuerst definieren.

Anschließend gehen wir der Frage nach, mit welchen Kombinationen der Verknüpfungszeichen $\top$, $\bot$, $\neg$, $\wedge$, $\vee$, $\rightarrow$ man bereits alle Formeln äquivalent beschreiben kann. Wir werden eine Kombination finden, die uns später das Beweisen leichter machen wird.

2.1 Äquivalente Formeln

Abkürzende Schreibweise: wir schreiben auch $A, B, C, \ldots$ für $A_0, A_1, A_2, \ldots$.

Die Formeln $(A \wedge B)$ und $\neg(\neg A \vee \neg B)$ werden von den gleichen Belegungen erfüllt.

- Die Belegung $\{A, B\}$ erfüllt beide Formeln.
- Die Belegungen $\emptyset$, $\{A\}$ und $\{B\}$ erfüllen beide Formeln nicht.
- Jede andere Belegung entspricht für die Atome A und B einer der obigen Belegungen.

Definition 2.1 ((semantische) Äquivalenz von Formeln)

Seien α und β Formeln.
Die Relation $\equiv$ zwischen Formeln ist wie folgt definiert:
$\alpha \equiv \beta$ genau dann, wenn
für jede Belegung $\mathcal{B}$ gilt: $\mathcal{B} \models \alpha$ genau dann, wenn $\mathcal{B} \models \beta$.

Die Aussage „$\alpha \equiv \beta$“ spricht man „α ist äquivalent (zu) β“ aus.
$\equiv$ ist eine Äquivalenzrelation.

Äquivalenzen, die wir noch brauchen werden

Lemma 2.2

Für jede Formel α gilt: $\neg\alpha \equiv \alpha \to \bot$.

Beweis:

Sei α eine beliebige Formel.

Zu zeigen ist:

für jede Belegung $\mathcal{B}$ gilt: $\mathcal{B} \models \neg\alpha$ gdw. $\mathcal{B} \models \alpha \to \bot$.

Sei $\mathcal{B}$ eine beliebige Belegung.

Dann gilt:

$$\begin{array}{llll} \mathcal{B} \models \neg\alpha & \text{gdw.} & \mathcal{B} \not\models \alpha & \text{(Semantik von } \neg\text{)} \\ & \text{gdw.} & \mathcal{B} \not\models \alpha \text{ oder } \mathcal{B} \models \bot & \text{(da } \mathcal{B} \not\models \bot\text{)} \\ & \text{gdw.} & \mathcal{B} \models \alpha \to \bot & \text{(Semantik von } \to\text{)} \end{array}$$

Damit ist „$\mathcal{B} \models \neg\alpha$ gdw. $\mathcal{B} \models \alpha \to \bot$“ gezeigt. ✓

Da $\top \equiv \neg\bot$, folgt damit $\top \equiv \bot \to \bot$.

Lemma 2.3

Für alle Formeln α und β gilt: $\alpha \vee \beta \equiv (\alpha \to \bot) \to \beta$.

Beweis:

Seien α und β beliebige Formeln. Zu zeigen ist:

für jede Belegung $\mathcal{B}$ gilt: $\mathcal{B} \models \alpha \vee \beta$ gdw. $\mathcal{B} \models (\alpha \to \bot) \to \beta$.

Sei $\mathcal{B}$ eine beliebige Belegung.

Dann gilt:

$$
\begin{array}{lll}
\mathcal{B} \models \alpha \vee \beta & \text{gdw.} \quad \mathcal{B} \models \alpha \text{ oder } \mathcal{B} \models \beta & \text{(Semantik von } \vee) \\
& \text{gdw.} \quad \mathcal{B} \not\models \neg\alpha \text{ oder } \mathcal{B} \models \beta & \text{(Semantik von } \neg) \\
& \text{gdw.} \quad \mathcal{B} \not\models \alpha \to \bot \text{ oder } \mathcal{B} \models \beta & \text{(Lemma (2.2))} \\
& \text{gdw.} \quad \mathcal{B} \models (\alpha \to \bot) \to \beta & \text{(Semantik von } \to)
\end{array}
$$

Damit ist „$\mathcal{B} \models \alpha \vee \beta$ gdw. $\mathcal{B} \models (\alpha \to \bot) \to \beta$“ gezeigt. ✓

Lemma 2.4

Für alle Formeln α und β gilt: $\alpha \wedge \beta \equiv (\alpha \to (\beta \to \bot)) \to \bot$.

Beweis:

Seien α und β beliebige Formeln. Zu zeigen ist:

für jede Belegung $\mathcal{B}$ gilt: $\mathcal{B} \models \alpha \wedge \beta$ gdw. $\mathcal{B} \models (\alpha \to (\beta \to \bot)) \to \bot$.

Sei $\mathcal{B}$ eine beliebige Belegung.

Dann gilt:

$\mathcal{B} \models \alpha \wedge \beta$	gdw.	$\mathcal{B} \models \alpha$ und $\mathcal{B} \models \beta$	(Semantik von $\wedge$)
	gdw.	„$\mathcal{B} \not\models \alpha$ oder $\mathcal{B} \not\models \beta$“ ist falsch	(ugs.)
	gdw.	„$\mathcal{B} \not\models \alpha$ oder $\mathcal{B} \models \neg\beta$“ ist falsch	(Semantik von $\neg$)
	gdw.	„$\mathcal{B} \not\models \alpha$ oder $\mathcal{B} \models \beta \to \bot$“ ist falsch	(Lemma (2.2))
	gdw.	„$\mathcal{B} \models \alpha \to (\beta \to \bot)$“ ist falsch	(Semantik von $\to$)
	gdw.	$\mathcal{B} \not\models \alpha \to (\beta \to \bot)$	(Schreibweise)
	gdw.	$\mathcal{B} \models \neg(\alpha \to (\beta \to \bot))$	(Semantik von $\neg$)
	gdw.	$\mathcal{B} \models (\alpha \to (\beta \to \bot)) \to \bot$	(Lemma (2.2))

Damit ist „$\mathcal{B} \models \alpha \wedge \beta$ gdw. $\mathcal{B} \models (\alpha \to (\beta \to \bot)) \to \bot$“ gezeigt. ✓

Lemma 2.5 (äquivalente Ersetzung von Teilformeln)

Seien α, β, α' und β' Formeln, und es gelte $\alpha \equiv \alpha'$ und $\beta \equiv \beta'$.
Dann gilt:

1. $\neg\alpha \equiv \neg\alpha'$

2. $\alpha \wedge \beta \equiv \alpha' \wedge \beta'$

3. $\alpha \vee \beta \equiv \alpha' \vee \beta'$

4. $\alpha \to \beta \equiv \alpha' \to \beta'$

Beweis:
1. Seien α und α' Formeln mit $\alpha \equiv \alpha'$.
Zu zeigen ist: für jede Belegung $\mathcal{B}$ gilt $\mathcal{B} \models \neg\alpha$ gdw. $\mathcal{B} \models \neg\alpha'$.
Sei $\mathcal{B}$ eine Belegung. Dann gilt:

$$\begin{array}{llll} \mathcal{B} \models \neg\alpha & \text{gdw.} & \mathcal{B} \not\models \alpha & (\text{Semantik von } \neg) \\ & \text{gdw.} & \mathcal{B} \not\models \alpha' & (\text{da } \alpha \equiv \alpha', \text{ d.h. } \mathcal{B} \models \alpha \text{ gdw. } \mathcal{B} \models \alpha') \\ & \text{gdw.} & \mathcal{B} \models \neg\alpha' & (\text{Semantik von } \neg) \end{array}$$

2.-4.: geht entsprechend. ✓

2.2 Adäquate Verknüpfungszeichen

Definition 2.6 (adäquate Mengen von Verknüpfungszeichen)

Eine Menge $M \subseteq \{\top, \bot, \neg, \wedge, \vee, \rightarrow, \ldots\}$ von Verknüpfungszeichen heißt adäquat, falls es für jede Formel eine äquivalente Formel gibt, die nur aus Atomen sowie Verknüpfungszeichen aus M besteht.

Beispiel:

Da $((\neg\alpha) \rightarrow \beta) \equiv (\alpha \vee \beta)$,
kann man alle Formeln äquivalent durch Formeln ohne $\vee$ ausdrücken.

Also ist $\{\top, \bot, \neg, \wedge, \rightarrow\}$ adäquat.

Lemma 2.7 ($\{\perp, \to\}$ ist adäquat)

$\{\perp, \to\}$ *ist eine adäquate Menge von Verknüpfungszeichen.*

D.h. für jede aussagenlogische Formel φ gibt es eine äquivalente Formel φ', die nur aus Atomen, $\perp$ und $\to$ besteht.

Der Beweis soll mittels Induktion über den Formelaufbau von φ geführt werden.
Was ist zu zeigen?

Induktionsanfang:
Für jede aussagenlogische Formel φ, die $\top$, $\perp$ oder ein Atom ist,
gibt es eine äquivalente Formel φ', die nur aus Atomen, $\perp$ und $\to$ besteht.

Induktionsschritt:
Für alle aussagenlogischen Formeln α und β gilt:
wenn α und β äquivalente Formeln besitzen, die nur aus Atomen, $\perp$ und $\to$ bestehen,
dann gibt es auch für $\neg\alpha$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$ und $(\alpha \to \beta)$ äquivalente Formeln,
die nur aus Atomen, $\perp$ und $\to$ bestehen.

Der Induktionsschritt wird in zwei Teile aufgeteilt:

Induktionsvoraussetzung:
α und β seien beliebige Formeln mit äquivalenten Formeln $\alpha' \equiv \alpha$ bzw. $\beta' \equiv \beta$,
die nur aus Atomen, $\perp$ und $\to$ bestehen.

Induktionsschluss – für α und β aus der Induktionsvoraussetzung ist zu zeigen:
$\neg\alpha$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$ und $(\alpha \to \beta)$ besitzen äquivalente Formeln,
die nur aus Atomen, $\perp$ und $\to$ bestehen.

Beweis: mittels Induktion über den Formelaufbau der Formel φ.

Induktionsanfang (IA):

Zu zeigen ist:
Für jede aussagenlogische Formel φ, die $\bot$, $\top$ oder ein Atom ist,
gibt es eine äquivalente Formel φ', die nur aus Atomen, $\bot$ und $\to$ besteht.

Fall 1: $\varphi = \top$. Dann ist $\varphi' = (\bot \to \bot)$ äquivalent zu φ,
und φ' besteht nur aus Atomen, $\bot$ und $\to$.

Fall 2: $\varphi = \bot$. Dann ist $\varphi' = \bot$ äquivalent zu φ,
und φ' besteht nur aus Atomen, $\bot$ und $\to$.

Fall 3: $\varphi = A_i$. Dann ist $\varphi' = A_i$ äquivalent zu φ,
und φ' besteht nur aus Atomen, $\bot$ und $\to$.

Induktionsvoraussetzung (IV):
α und β besitzen äquivalente Formeln α' bzw. β',
die nur aus Atomen, $\perp$ und $\rightarrow$ bestehen.

Induktionsschluss (IS):

Zu zeigen ist: $\neg\alpha$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$ und $(\alpha \rightarrow \beta)$ besitzen äquivalente Formeln, die nur aus Atomen, $\perp$ und $\rightarrow$ bestehen.

Fall 1: $\varphi = \neg\alpha$.
Es gilt:

$$\begin{aligned} \neg\alpha &\equiv \neg\,\alpha' && \text{(IV und Lemma (2.5))} \\ &\equiv \alpha' \rightarrow \perp && \text{(Lemma (2.2))} \end{aligned}$$

Also ist $\varphi' = (\alpha' \rightarrow \perp)$ äquivalent zu φ.
Da in α' laut IV nur Atome, $\perp$ und $\rightarrow$ vorkommen,
kommen in φ' nur Atome, $\perp$ und $\rightarrow$ vor.

Fall 2: $\varphi = (\alpha \wedge \beta)$.

Es gilt:

$$\begin{aligned} (\alpha \wedge \beta) &\equiv (\alpha' \wedge \beta') && \text{(IV und Lemma (2.5))} \\ &\equiv ((\alpha' \rightarrow (\beta' \rightarrow \perp)) \rightarrow \perp) && \text{(Lemma (2.3))} \end{aligned}$$

Also ist $\varphi' = ((\alpha' \to (\beta' \to \bot)) \to \bot)$ äquivalent zu φ.
Da in α' und β' laut IV nur Atome, $\bot$ und $\to$ vorkommen,
kommen in φ' nur Atome, $\bot$ und $\to$ vor.

Fall 3: $\varphi = (\alpha \vee \beta)$.
Es gilt: $(\alpha \vee \beta)$

$$\begin{aligned} &\equiv (\alpha' \vee \beta') && \text{(IV und Lemma (2.5))} \\ &\equiv ((\alpha' \to \bot) \to \beta') && \text{(Lemma (2.4))} \end{aligned}$$

Dann ist $\varphi' = ((\alpha' \to \bot) \to \beta')$ äquivalent zu φ.
Da in α' und β' laut IV nur Atome, $\bot$ und $\to$ vorkommen,
kommen in φ' nur Atome, $\bot$ und $\to$ vor.

Fall 4: $\varphi = (\alpha \to \beta)$.
Dann ist $\varphi' = (\alpha' \to \beta')$ äquivalent zu φ (IV und Lemma (2.5)),
und in φ' kommen gemäß IV nur Atome, $\bot$ und $\to$ vor.

✓

$\{\perp, \to\}$ ist nicht die einzige adäquate Menge.

Satz 2.8 (Adäquate Mengen von Verknüpfungszeichen)

Die folgenden Mengen von Verknüpfungszeichen sind adäquat.

1. $\{\perp, \to\}$
2. $\{\neg, \wedge\}$
3. $\{\neg, \vee\}$
4. $\{\neg, \to\}$

Die Beweise für die anderen Mengen gehen ähnlich wie der Beweis von Lemma (2.7).

Was haben wir in Vorlesung 2 gelernt?

- Wir kennen die Relation $\equiv$ der Äquivalenz von Formeln.
- Wir wissen, wie man die Äquivalenz von Formeln beweisen kann.
- Wir wissen, dass $\{\bot, \to\}$ eine adäquate Mengen von Verknüpfungszeichen ist und können das mittels Induktion über den Formelaufbau beweisen.
- Wir kennen weitere Mengen adäquater Verknüpfungszeichen.

1.3–5
Zwei aussagenlogische Beweis-Kalküle

1.3–5 Aussagenlogische Beweis-Kalküle

Wir wollen überprüfen, ob eine Folgerung $\Gamma \Vdash \alpha$ korrekt ist
— für den Fall $\Gamma = \emptyset$ heißt das: α ist gültig.
Wir wollen also eine semantische Eigenschaft von Formeln überprüfen.
Ein *Beweis-Kalkül* liefert ein Verfahren dafür.
Diese Verfahren arbeiten rein syntaktisch – d.h. sie argumentieren nur über die Struktur der Formeln.
Wir werden zwei beispielhafte Kalküle kennenlernen:

- Tableau-Kalkül
  (die Formel wird auseinandergenommen, um sie zu analysieren)
- Frege-Kalkül
  (die Formel wird zusammengesetzt, um sie zu analysieren)

Da Beweis-Kalküle syntaktisch arbeiten, der Begriff der Gültigkeit jedoch semantisch ist, ist nicht offensichtlich, dass die beiden Kalküle genau die gültigen Formeln herausfinden. Deshalb muss das abschließend bewiesen werden – und das ist die Hauptarbeit in diesen Vorlesungen.

Vorlesung 3: Ein Tableau-Kalkül

3.1 Einführende Beispiele (1)

$$\neg(A \wedge B) \wedge \neg(\neg A \wedge \neg B)$$

Ein (semantisches) Tableau ist ein Baum,
dessen Knoten mit Formeln markiert sind.

Man beginnt mit einem Baum, der nur aus einer Wurzel besteht.
Die Knoten werden nach gegebenen Regeln expandiert und die Markierungen dadurch in ihre Bestandteile zerlegt „bis auf die Literale“.

3.1 Einführende Beispiele (1)

$\neg(A \wedge B) \wedge \neg(\neg A \wedge \neg B)$

Expansionsregeln:

$\alpha \wedge \beta$:

$$\begin{array}{c} \bullet \\ \downarrow \\ \alpha \\ \downarrow \\ \beta \end{array}$$

Ein (semantisches) Tableau ist ein Baum,
dessen Knoten mit Formeln markiert sind.

Man beginnt mit einem Baum, der nur aus einer Wurzel besteht.
Die Knoten werden nach gegebenen Regeln expandiert und die Markierungen dadurch in ihre Bestandteile zerlegt „bis auf die Literale“.

3.1 Einführende Beispiele (1)

$$\neg(A \wedge B) \wedge \neg(\neg A \wedge \neg B)$$
$$\downarrow$$
$$\neg(A \wedge B)$$
$$\downarrow$$
$$\neg(\neg A \wedge \neg B)$$

Expansionsregeln:

$\alpha \wedge \beta$:

$$\bullet$$
$$\downarrow$$
$$\alpha$$
$$\downarrow$$
$$\beta$$

Ein (semantisches) Tableau ist ein Baum,
dessen Knoten mit Formeln markiert sind.

Man beginnt mit einem Baum, der nur aus einer Wurzel besteht.
Die Knoten werden nach gegebenen Regeln expandiert und die Markierungen dadurch in ihre Bestandteile zerlegt „bis auf die Literale“.

3.1 Einführende Beispiele (1)

$$\begin{array}{c} \neg(A \wedge B) \wedge \neg(\neg A \wedge \neg B) \\ \downarrow \\ \neg(A \wedge B) \\ \downarrow \\ \neg(\neg A \wedge \neg B) \end{array}$$

Expansionsregeln:

$\alpha \wedge \beta$:

$$\begin{tikzcd} \bullet \arrow[d] \\ \alpha \arrow[d] \\ \beta \end{tikzcd}$$

$\neg(\alpha \wedge \beta)$:

$$\begin{tikzcd} & \bullet \arrow[dl] \arrow[dr] & \\ \neg\alpha & & \neg\beta \end{tikzcd}$$

Ein (semantisches) Tableau ist ein Baum,
dessen Knoten mit Formeln markiert sind.

Man beginnt mit einem Baum, der nur aus einer Wurzel besteht.
Die Knoten werden nach gegebenen Regeln expandiert und die Markierungen dadurch in ihre Bestandteile zerlegt „bis auf die Literale“.

3.1 Einführende Beispiele (1)

$$
\begin{tikzcd}[column sep=small]
 & \neg(A \land B) \land \neg(\neg A \land \neg B) \arrow[d] & \\
 & \neg(A \land B) \arrow[d] & \\
 & \neg(\neg A \land \neg B) \arrow[dl] \arrow[dr] & \\
\neg A & & \neg B
\end{tikzcd}
$$

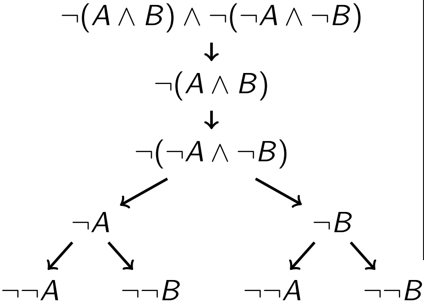
Expansionsregeln:

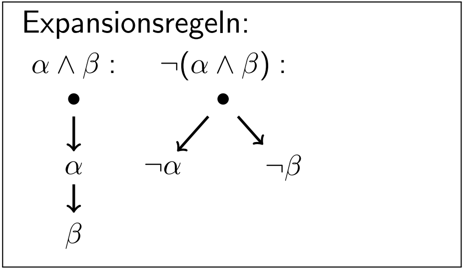
$$
\alpha \land \beta : \quad
\begin{tikzcd}
\bullet \arrow[d] \\
\alpha \arrow[d] \\
\beta
\end{tikzcd}
\qquad
\neg(\alpha \land \beta) : \quad
\begin{tikzcd}[column sep=small]
 & \bullet \arrow[dl] \arrow[dr] & \\
\neg\alpha & & \neg\beta
\end{tikzcd}
$$

Ein (semantisches) Tableau ist ein Baum,
dessen Knoten mit Formeln markiert sind.

Man beginnt mit einem Baum, der nur aus einer Wurzel besteht.
Die Knoten werden nach gegebenen Regeln expandiert und die Markierungen dadurch in ihre Bestandteile zerlegt „bis auf die Literale“.

3.1 Einführende Beispiele (1)

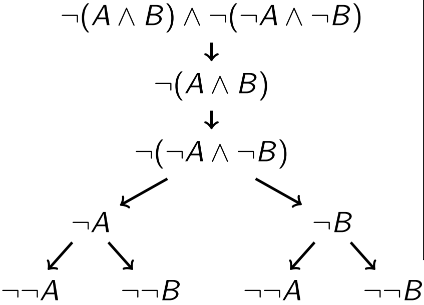


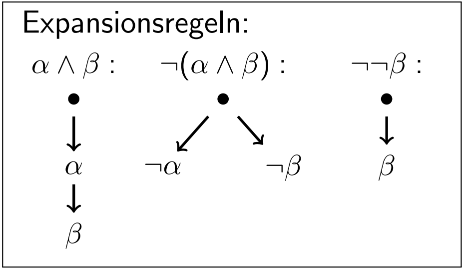


Ein (semantisches) Tableau ist ein Baum,
dessen Knoten mit Formeln markiert sind.

Man beginnt mit einem Baum, der nur aus einer Wurzel besteht.
Die Knoten werden nach gegebenen Regeln expandiert und die Markierungen dadurch in ihre Bestandteile zerlegt „bis auf die Literale“.

3.1 Einführende Beispiele (1)

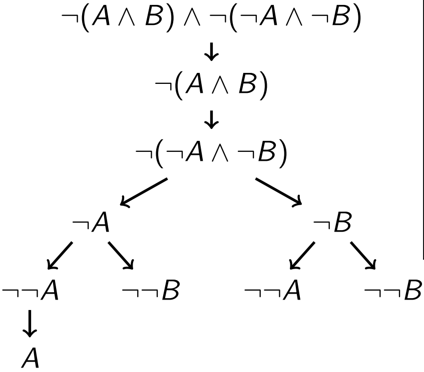


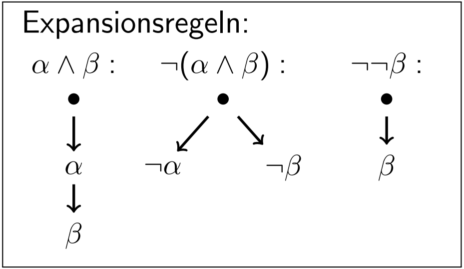


Ein (semantisches) Tableau ist ein Baum,
dessen Knoten mit Formeln markiert sind.

Man beginnt mit einem Baum, der nur aus einer Wurzel besteht.
Die Knoten werden nach gegebenen Regeln expandiert und die Markierungen dadurch in ihre Bestandteile zerlegt „bis auf die Literale“.

3.1 Einführende Beispiele (1)

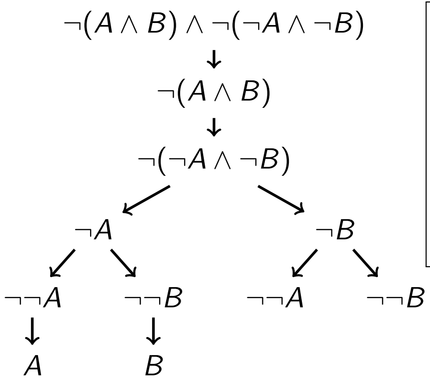


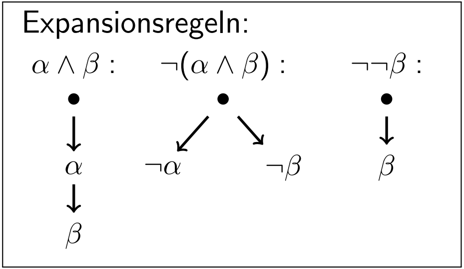


Ein (semantisches) Tableau ist ein Baum, dessen Knoten mit Formeln markiert sind.

Man beginnt mit einem Baum, der nur aus einer Wurzel besteht.
Die Knoten werden nach gegebenen Regeln expandiert und die Markierungen dadurch in ihre Bestandteile zerlegt „bis auf die Literale“.

3.1 Einführende Beispiele (1)

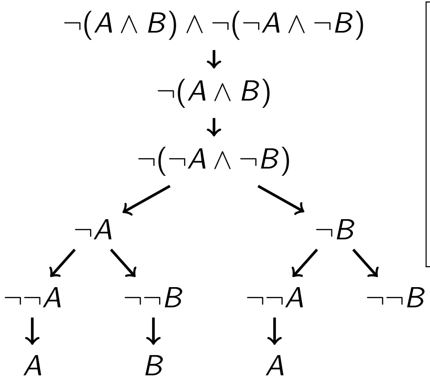


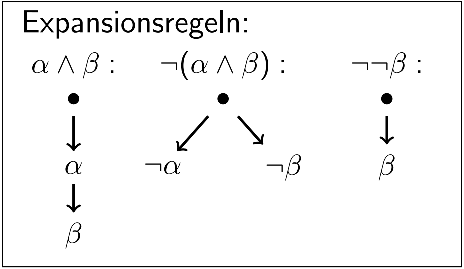


Ein (semantisches) Tableau ist ein Baum,
dessen Knoten mit Formeln markiert sind.

Man beginnt mit einem Baum, der nur aus einer Wurzel besteht.
Die Knoten werden nach gegebenen Regeln expandiert und die Markierungen dadurch in ihre Bestandteile zerlegt „bis auf die Literale“.

3.1 Einführende Beispiele (1)

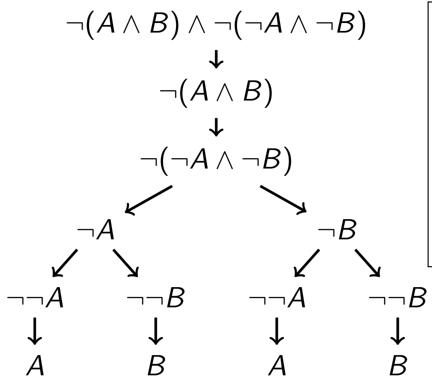


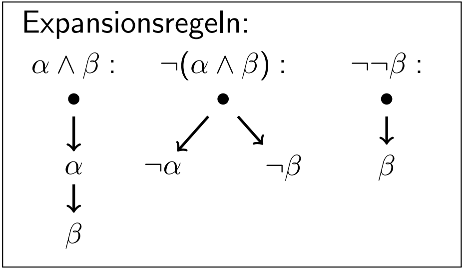


Ein (semantisches) Tableau ist ein Baum,
dessen Knoten mit Formeln markiert sind.

Man beginnt mit einem Baum, der nur aus einer Wurzel besteht.
Die Knoten werden nach gegebenen Regeln expandiert und die Markierungen dadurch in ihre Bestandteile zerlegt „bis auf die Literale".

3.1 Einführende Beispiele (1)





Wenn alle Knoten expandiert sind,
dann gibt es für jede erfüllende Belegung der Wurzel-Formel einen Pfad,
auf dem die Belegung alle Formeln erfüllt,
und jeder nicht-widersprüchliche Pfad bestimmt
erfüllende Belegungen der Wurzel-Formel.

Erfüllende Belegungen sind hier: $\{A\}$ und $\{B\}$.

Einführende Beispiele (2)

$\neg(A \wedge \neg(A \wedge \neg B))$ wird expandiert zu $\quad \neg(A \wedge \neg(A \wedge \neg B))$

Expansionsregeln:

$$\begin{array}{ccc}
\alpha \wedge \beta : & \neg(\alpha \wedge \beta) : & \neg\neg\beta : \\
\bullet & \bullet & \bullet \\
\downarrow & \swarrow \quad \searrow & \downarrow \\
\alpha & \neg\alpha \qquad \neg\beta & \beta \\
\downarrow & & \\
\beta & &
\end{array}$$

Einführende Beispiele (2)

$\neg(A \wedge \neg(A \wedge \neg B))$ wird expandiert zu

$$\begin{array}{ccc} & \neg(A \wedge \neg(A \wedge \neg B)) & \\ \swarrow & & \searrow \\ \neg A & & \neg\neg(A \wedge \neg B) \end{array}$$

Expansionsregeln:

$\alpha \wedge \beta$:
$$\begin{array}{c} \bullet \\ \downarrow \\ \alpha \\ \downarrow \\ \beta \end{array}$$

$\neg(\alpha \wedge \beta)$:
$$\begin{array}{ccc} & \bullet & \\ \swarrow & & \searrow \\ \neg\alpha & & \neg\beta \end{array}$$

$\neg\neg\beta$:
$$\begin{array}{c} \bullet \\ \downarrow \\ \beta \end{array}$$

Einführende Beispiele (2)

$\neg(A \wedge \neg(A \wedge \neg B))$ wird expandiert zu

$$\begin{tikzcd}[column sep=small] & \neg(A \wedge \neg(A \wedge \neg B)) \arrow[dl] \arrow[dr] & \\ \neg A & & \neg\neg(A \wedge \neg B) \arrow[d] \\ & & A \wedge \neg B \end{tikzcd}$$

Expansionsregeln:

$\alpha \wedge \beta$:

$$\begin{tikzcd} \bullet \arrow[d] \\ \alpha \arrow[d] \\ \beta \end{tikzcd}$$

$\neg(\alpha \wedge \beta)$:

$$\begin{tikzcd}[column sep=small] & \bullet \arrow[dl] \arrow[dr] & \\ \neg\alpha & & \neg\beta \end{tikzcd}$$

$\neg\neg\beta$:

$$\begin{tikzcd} \bullet \arrow[d] \\ \beta \end{tikzcd}$$

Einführende Beispiele (2)

$\neg(A \wedge \neg(A \wedge \neg B))$ wird expandiert zu

$$\begin{tikzcd}[column sep=small]
 & \neg(A \wedge \neg(A \wedge \neg B)) \arrow[dl] \arrow[dr] & \\
\neg A & & \neg\neg(A \wedge \neg B) \arrow[d] \\
 & & A \wedge \neg B \arrow[d] \\
 & & A \arrow[d] \\
 & & \neg B
\end{tikzcd}$$

Expansionsregeln:

$\alpha \wedge \beta$:
$$\begin{tikzcd}
\bullet \arrow[d] \\
\alpha \arrow[d] \\
\beta
\end{tikzcd}$$

$\neg(\alpha \wedge \beta)$:
$$\begin{tikzcd}[column sep=small]
 & \bullet \arrow[dl] \arrow[dr] & \\
\neg\alpha & & \neg\beta
\end{tikzcd}$$

$\neg\neg\beta$:
$$\begin{tikzcd}
\bullet \arrow[d] \\
\beta
\end{tikzcd}$$

Einführende Beispiele (2)

$\neg(A \land \neg(A \land \neg B))$ wird expandiert zu

$$
\begin{array}{ccc}
 & \neg(A \land \neg(A \land \neg B)) & \\
\swarrow & & \searrow \\
\neg A & & \neg\neg(A \land \neg B) \\
 & & \downarrow \\
 & & A \land \neg B \\
 & & \downarrow \\
 & & A \\
 & & \downarrow \\
 & & \neg B
\end{array}
$$

Expansionsregeln:

$$
\begin{array}{c}
\alpha \land \beta : \\
\bullet \\
\downarrow \\
\alpha \\
\downarrow \\
\beta
\end{array}
\qquad
\begin{array}{ccc}
 & \neg(\alpha \land \beta) : & \\
 & \bullet & \\
\swarrow & & \searrow \\
\neg\alpha & & \neg\beta
\end{array}
\qquad
\begin{array}{c}
\neg\neg\beta : \\
\bullet \\
\downarrow \\
\beta
\end{array}
$$

Erfüllende Belegungen: $\emptyset$, $\{A\}$ und $\{B\}$.

Einführende Beispiele (2)

$\neg(A \wedge \neg(A \wedge \neg B))$ wird expandiert zu

$$\begin{tikzcd}[column sep=small]
 & \neg(A \wedge \neg(A \wedge \neg B)) \arrow[dl] \arrow[dr] & \\
\neg A & & \neg\neg(A \wedge \neg B) \arrow[d] \\
 & & A \wedge \neg B \arrow[d] \\
 & & A \arrow[d] \\
 & & \neg B
\end{tikzcd}$$

Expansionsregeln:

$\alpha \wedge \beta$:

$$\begin{tikzcd}
\bullet \arrow[d] \\
\alpha \arrow[d] \\
\beta
\end{tikzcd}$$

$\neg(\alpha \wedge \beta)$:

$$\begin{tikzcd}
 & \bullet \arrow[dl] \arrow[dr] & \\
\neg\alpha & & \neg\beta
\end{tikzcd}$$

$\neg\neg\beta$:

$$\begin{tikzcd}
\bullet \arrow[d] \\
\beta
\end{tikzcd}$$

Erfüllende Belegungen: $\emptyset$, $\{A\}$ und $\{B\}$.

Disjunktive Normalform: $\neg A \vee (A \wedge \neg B)$

Einführende Beispiele (2)

$\neg(A \wedge \neg(A \wedge \neg B))$ wird expandiert zu

$$
\begin{tikzcd}[column sep=small]
 & \neg(A \wedge \neg(A \wedge \neg B)) \arrow[dl] \arrow[dr] & \\
\neg A & & \neg\neg(A \wedge \neg B) \arrow[d] \\
 & & A \wedge \neg B \arrow[d] \\
 & & A \arrow[d] \\
 & & \neg B
\end{tikzcd}
$$

Expansionsregeln:

$\alpha \wedge \beta$:

$$
\begin{tikzcd}
\bullet \arrow[d] \\
\alpha \arrow[d] \\
\beta
\end{tikzcd}
$$

$\neg(\alpha \wedge \beta)$:

$$
\begin{tikzcd}
 & \bullet \arrow[dl] \arrow[dr] & \\
\neg\alpha & & \neg\beta
\end{tikzcd}
$$

$\neg\neg\beta$:

$$
\begin{tikzcd}
\bullet \arrow[d] \\
\beta
\end{tikzcd}
$$

Erfüllende Belegungen: $\emptyset$, $\{A\}$ und $\{B\}$.

Disjunktive Normalform: $\neg A \vee (A \wedge \neg B) \equiv \neg A \vee \neg B$

3.2 Tableaux

Definition 3.1 (Tableau für Formeln aus Atomen, $\bot$, $\top$, $\neg$ und $\wedge$)

Sei Δ eine nicht-leere endliche Menge aussagenlogischer Formeln aus Atomen, $\bot$, $\top$, $\neg$, $\wedge$.

1. Ein Pfad, dessen Knoten mit Formeln aus Δ markiert sind, ist ein Tableau für Δ.
2. Sei T ein Tableau für Δ, und v sei ein Knoten in T mit Markierung ψ.
   Wenn ψ eine Formel der Form $\alpha \wedge \beta$, $\neg(\alpha \wedge \beta)$ oder $\neg\neg\alpha$ ist,
   dann kann v mit der entsprechenden Expansionsregel expandiert werden.

$\alpha \wedge \beta$:
$$\begin{tikzcd} \bullet \arrow[d] \\ \alpha \arrow[d] \\ \beta \end{tikzcd}$$

$\neg(\alpha \wedge \beta)$:
$$\begin{tikzcd} & \bullet \arrow[dl] \arrow[dr] & \\ \neg\alpha & & \neg\beta \end{tikzcd}$$

$\neg\neg\alpha$:
$$\begin{tikzcd} \bullet \arrow[d] \\ \alpha \end{tikzcd}$$

Expansion von v heißt:
für jeden Pfad durch T, auf dem v vorkommt:
hänge die durch die Expansionsregel für ψ bezeichneten Knoten an das Ende des Pfades an.
In der Expansionsregel steht $\bullet$ für den letzten Knoten im Pfad.

Durch Expansion von v entsteht ein (weiteres) Tableau für Δ.

Falls $\Delta = \{\varphi\}$, schreiben wir vereinfachend auch „Tableau für φ“.

Systematischer Aufbau eines Tableaus für

$$\{\neg((A \wedge \neg C) \wedge (B \wedge \neg C)) \wedge (\neg(\neg A \wedge \neg B) \wedge \neg C\,)\}$$

Jeder Knoten wird einmal expandiert.

Der nächste zu expandierende Knoten ist

der erste (gemäß pre-order) nicht-expandierte Knoten des Baumes.

Systematischer Aufbau eines Tableaus

$$\neg\big((A \wedge \neg C) \wedge (B \wedge \neg C)\big) \wedge \big(\neg(\neg A \wedge \neg B) \wedge \neg C\,\big)$$
$$\downarrow$$
$$\neg((A \wedge \neg C) \wedge (B \wedge \neg C))$$
$$\downarrow$$
$$\neg(\neg A \wedge \neg B) \wedge \neg C$$

Jeder Knoten wird einmal expandiert.

Der nächste zu expandierende Knoten ist

der erste (gemäß pre-order) nicht-expandierte Knoten des Baumes.

Systematischer Aufbau eines Tableaus

$$\begin{tikzcd}[column sep=tiny]
 & \neg\big((A \wedge \neg C) \wedge (B \wedge \neg C)\big) \wedge \big(\neg(\neg A \wedge \neg B) \wedge \neg C\,\big) \arrow[d] & \\
 & \neg((A \wedge \neg C) \wedge (B \wedge \neg C)) \arrow[d] & \\
 & \neg(\neg A \wedge \neg B) \wedge \neg C \arrow[dl] \arrow[dr] & \\
\neg(A \wedge \neg C) & & \neg(B \wedge \neg C)
\end{tikzcd}$$

Jeder Knoten wird einmal expandiert.

Der nächste zu expandierende Knoten ist

der erste (gemäß pre-order) nicht-expandierte Knoten des Baumes.

Systematischer Aufbau eines Tableaus

$$
\begin{tikzcd}[column sep=small]
& \neg\big((A \land \neg C) \land (B \land \neg C)\big) \land \big(\neg(\neg A \land \neg B) \land \neg C\big) \arrow[d] & \\
& \neg((A \land \neg C) \land (B \land \neg C)) \arrow[d] & \\
& \neg(\neg A \land \neg B) \land \neg C \arrow[dl] \arrow[dr] & \\
\neg(A \land \neg C) \arrow[d] & & \neg(B \land \neg C) \arrow[d] \\
\neg(\neg A \land \neg B) \arrow[d] & & \neg(\neg A \land \neg B) \arrow[d] \\
\neg C & & \neg C
\end{tikzcd}
$$

Jeder Knoten wird einmal expandiert.

Der nächste zu expandierende Knoten ist

der erste (gemäß pre-order) nicht-expandierte Knoten des Baumes.

Systematischer Aufbau eines Tableaus

$$
\begin{tikzcd}[column sep=small]
 & & \neg((A \wedge \neg C) \wedge (B \wedge \neg C)) \wedge (\neg(\neg A \wedge \neg B) \wedge \neg C) \arrow[d] & & \\
 & & \neg((A \wedge \neg C) \wedge (B \wedge \neg C)) \arrow[d] & & \\
 & & \neg(\neg A \wedge \neg B) \wedge \neg C \arrow[dl] \arrow[dr] & & \\
 & \neg(A \wedge \neg C) \arrow[d] & & \neg(B \wedge \neg C) \arrow[d] & \\
 & \neg(\neg A \wedge \neg B) \arrow[d] & & \neg(\neg A \wedge \neg B) \arrow[d] & \\
 & \neg C \arrow[dl] \arrow[dr] & & \neg C & \\
 \neg A & & \neg\neg C & & \\
 & & \times & &
\end{tikzcd}
$$

Jeder Knoten wird einmal expandiert.

Der nächste zu expandierende Knoten ist

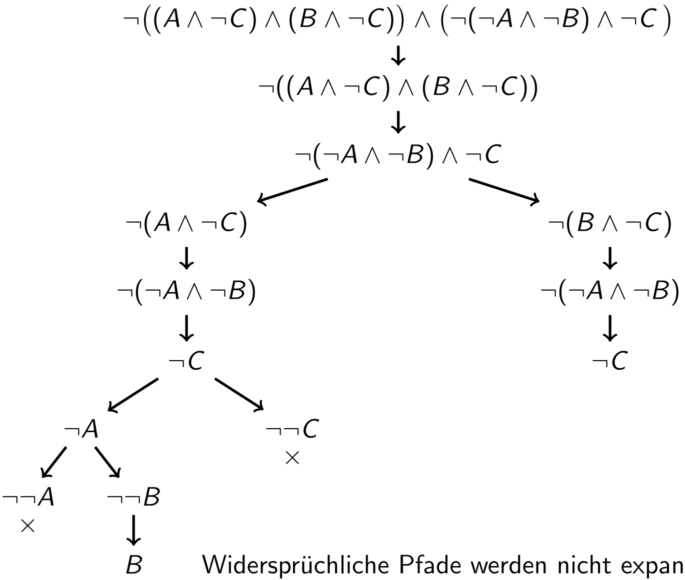
der erste (gemäß pre-order) nicht-expandierte Knoten des Baumes.

Systematischer Aufbau eines Tableaus

$$
\begin{tikzcd}[column sep=small]
 & & & & & \neg\big((A \wedge \neg C) \wedge (B \wedge \neg C)\big) \wedge \big(\neg(\neg A \wedge \neg B) \wedge \neg C\,\big) \arrow[d] & \\
 & & & & & \neg((A \wedge \neg C) \wedge (B \wedge \neg C)) \arrow[d] & \\
 & & & & & \neg(\neg A \wedge \neg B) \wedge \neg C \arrow[dll] \arrow[dr] & \\
 & & & \neg(A \wedge \neg C) \arrow[d] & & & \neg(B \wedge \neg C) \arrow[d] \\
 & & & \neg(\neg A \wedge \neg B) \arrow[d] & & & \neg(\neg A \wedge \neg B) \arrow[d] \\
 & & & \neg C \arrow[dll] \arrow[dr] & & & \neg C \\
 & \neg A \arrow[dl] \arrow[dr] & & & \neg\neg C & & \\
\neg\neg A & & \neg\neg B & & \times & & \\
\times & & & & & &
\end{tikzcd}
$$

Widersprüchliche Pfade werden nicht expandiert.

Systematischer Aufbau eines Tableaus



Widersprüchliche Pfade werden nicht expandiert.

Systematischer Aufbau eines Tableaus

- $\neg\big((A \wedge \neg C) \wedge (B \wedge \neg C)\big) \wedge \big(\neg(\neg A \wedge \neg B) \wedge \neg C\big)$
  - $\downarrow$ $\neg((A \wedge \neg C) \wedge (B \wedge \neg C))$
    - $\downarrow$ $\neg(\neg A \wedge \neg B) \wedge \neg C$
      - $\neg(A \wedge \neg C)$
        - $\downarrow$ $\neg(\neg A \wedge \neg B)$
          - $\downarrow$ $\neg C$
            - $\neg A$
              - $\neg\neg A$ $\times$
              - $\neg\neg B$
                - $\downarrow$ B
            - $\neg\neg C$ $\times$
      - $\neg(B \wedge \neg C)$
        - $\downarrow$ $\neg(\neg A \wedge \neg B)$
          - $\downarrow$ $\neg C$
            - $\neg B$
            - $\neg\neg C$ $\times$

Widersprüchliche Pfade werden nicht expandiert.

Systematischer Aufbau eines Tableaus

$$\neg\big((A \wedge \neg C) \wedge (B \wedge \neg C)\big) \wedge \big(\neg(\neg A \wedge \neg B) \wedge \neg C\big)$$

$$\downarrow$$

$$\neg((A \wedge \neg C) \wedge (B \wedge \neg C))$$

$$\downarrow$$

$$\neg(\neg A \wedge \neg B) \wedge \neg C$$

- $\neg(A \wedge \neg C)$
  - $\downarrow$ $\neg(\neg A \wedge \neg B)$
    - $\downarrow$ $\neg C$
      - $\neg A$
        - $\neg\neg A$ $\times$
        - $\neg\neg B$ $\downarrow$ B
      - $\neg\neg C$ $\times$
- $\neg(B \wedge \neg C)$
  - $\downarrow$ $\neg(\neg A \wedge \neg B)$
    - $\downarrow$ $\neg C$
      - $\neg B$
        - A
        - B $\times$
      - $\neg\neg C$ $\times$

Widersprüchliche Pfade werden nicht expandiert.

Systematischer Aufbau eines Tableaus

- $\neg\big((A \land \neg C) \land (B \land \neg C)\big) \land \big(\neg(\neg A \land \neg B) \land \neg C\big)$
  - $\neg((A \land \neg C) \land (B \land \neg C))$
    - $\neg(\neg A \land \neg B) \land \neg C$
      - $\neg(A \land \neg C)$
        - $\neg(\neg A \land \neg B)$
          - $\neg C$
            - $\neg A$
              - $\neg\neg A$ $\times$
              - $\neg\neg B$
                - B
            - $\neg\neg C$ $\times$
      - $\neg(B \land \neg C)$
        - $\neg(\neg A \land \neg B)$
          - $\neg C$
            - $\neg B$
              - A
              - B $\times$
            - $\neg\neg C$ $\times$

Erfüllende Belegungen: $\{B\}$ und $\{A\}$.

Fast systematischer Aufbau eines Tableaus für

$$\{\ \neg(A \wedge \neg B) \wedge \neg(B \wedge \neg C), \neg\neg(A \wedge \neg C)\ \}$$

Fast systematischer Aufbau eines Tableaus

$$\neg(A \wedge \neg B) \wedge \neg(B \wedge \neg C)$$

$$\downarrow$$

$$\neg\neg(A \wedge \neg C)$$

Fast systematischer Aufbau eines Tableaus

$$\neg(A \wedge \neg B) \wedge \neg(B \wedge \neg C)$$
$$\downarrow$$
$$\neg\neg(A \wedge \neg C)$$
$$\downarrow$$
$$\neg(A \wedge \neg B)$$
$$\downarrow$$
$$\neg(B \wedge \neg C)$$

Fast systematischer Aufbau eines Tableaus

$$
\begin{array}{c}
\neg(A \wedge \neg B) \wedge \neg(B \wedge \neg C) \\
\downarrow \\
\neg\neg(A \wedge \neg C) \\
\downarrow \\
\neg(A \wedge \neg B) \\
\downarrow \\
\neg(B \wedge \neg C) \\
\downarrow \\
A \wedge \neg C
\end{array}
$$

Fast systematischer Aufbau eines Tableaus

$$\neg(A \wedge \neg B) \wedge \neg(B \wedge \neg C)$$
$$\downarrow$$
$$\neg\neg(A \wedge \neg C)$$
$$\downarrow$$
$$\neg(A \wedge \neg B)$$
$$\downarrow$$
$$\neg(B \wedge \neg C)$$
$$\downarrow$$
$$A \wedge \neg C$$
$$\downarrow$$
$$A$$
$$\downarrow$$
$$\neg C$$

Fast systematischer Aufbau eines Tableaus

$$
\begin{tikzcd}[row sep=small]
& \neg(A \wedge \neg B) \wedge \neg(B \wedge \neg C) \arrow[d] & \\
& \neg\neg(A \wedge \neg C) \arrow[d] & \\
& \neg(A \wedge \neg B) \arrow[d] & \\
& \neg(B \wedge \neg C) \arrow[d] & \\
& A \wedge \neg C \arrow[d] & \\
& A \arrow[d] & \\
& \neg C \arrow[dl] \arrow[dr] & \\
\neg A & & \neg\neg B \\
\times & &
\end{tikzcd}
$$

Fast systematischer Aufbau eines Tableaus

$$
\begin{tikzcd}[row sep=small]
 & \neg(A \wedge \neg B) \wedge \neg(B \wedge \neg C) \arrow[d] & \\
 & \neg\neg(A \wedge \neg C) \arrow[d] & \\
 & \neg(A \wedge \neg B) \arrow[d] & \\
 & \neg(B \wedge \neg C) \arrow[d] & \\
 & A \wedge \neg C \arrow[d] & \\
 & A \arrow[d] & \\
 & \neg C \arrow[dl] \arrow[dr] & \\
\neg A & & \neg\neg B \arrow[d] \\
\times & & B
\end{tikzcd}
$$

Fast systematischer Aufbau eines Tableaus

$$
\begin{tikzcd}[column sep=small]
 & \neg(A \land \neg B) \land \neg(B \land \neg C) \arrow[d] & & & \\
 & \neg\neg(A \land \neg C) \arrow[d] & & & \\
 & \neg(A \land \neg B) \arrow[d] & & & \\
 & \neg(B \land \neg C) \arrow[d] & & & \\
 & A \land \neg C \arrow[d] & & & \\
 & A \arrow[d] & & & \\
 & \neg C \arrow[dl] \arrow[drr] & & & \\
\neg A & & & \neg\neg B \arrow[d] & \\
\times & & & B \arrow[dl] \arrow[dr] & \\
 & & \neg B & & \neg\neg C \\
 & & \times & & \times
\end{tikzcd}
$$

Definition 3.2 (Eigenschaften von Pfaden und Tableaux)

Ein Pfad durch ein Tableau heißt widersprüchlich, wenn
er $\bot$, $\neg\top$ oder zwei widersprüchliche Formeln α und $\neg\alpha$ enthält.

Ein Pfad durch ein Tableau heißt geschlossen,
wenn er nicht widersprüchlich ist und
jeder seiner expandierbaren Knoten expandiert wurde.

Ein Tableau heißt geschlossen,
wenn jeder Pfad durch das Tableau geschlossen oder widersprüchlich ist.
(Widersprüchliche Pfade werden nicht weiter expandiert.)

Ein Tableau heißt widersprüchlich,
wenn alle Pfade durch das Tableau widersprüchlich sind.

Uns interessiert,

ob eine Formelmenge ein widersprüchliches Tableau hat.

Wir werden sehen (Vollständigkeitssatz für den Tableau-Kalkül (5.12)):

- Formel α ist gültig genau dann,
  wenn $\{\neg\alpha\}$ ein widersprüchliches Tableau hat.

- $\Gamma \Vdash\!\models \alpha$ genau dann,
  wenn $\Gamma \cup \{\neg\alpha\}$ ein widersprüchliches Tableau hat.

In den folgenden drei Beispielen werden wir diesen Satz bereits benutzen (auch wenn wir ihn noch nicht bewiesen haben) ...

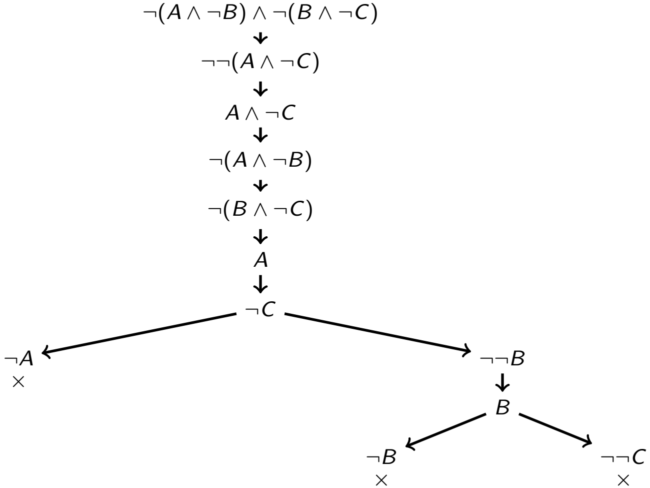
Bsp.: $\neg(A \wedge (B \wedge \neg A))$ ist gültig

Wir zeigen $\models \alpha$,
indem wir ein widersprüchliches Tableau für $\{\neg\alpha\}$ angeben:

$$\begin{array}{c}
\neg\neg(A \wedge (B \wedge \neg A)) \\
\downarrow \\
A \wedge (B \wedge \neg A) \\
\downarrow \\
A \\
\downarrow \\
B \wedge \neg A \\
\downarrow \\
B \\
\downarrow \\
\neg A \\
\times
\end{array}$$

Bsp.: $\neg(A \wedge \neg B) \wedge \neg(B \wedge \neg C) \Vdash\!\models \neg(A \wedge \neg C)$

Wir zeigen $\Gamma \Vdash\!\models \alpha$,
indem wir ein widersprüchliches Tableau für $\Gamma \cup \{\neg\alpha\}$ angeben:



Expansionsregeln für andere Verknüpfungszeichen

Man kann Tableau-Kalküle für Formelmengen mit „beliebigen“ Verknüpfungszeichen definieren.

Typ 1:

$$\begin{array}{cccc}
\alpha \wedge \beta : & \neg(\alpha \vee \beta) : & \neg(\alpha \rightarrow \beta) : & \neg\neg\alpha : \\
\bullet & \bullet & \bullet & \bullet \\
\downarrow & \downarrow & \downarrow & \downarrow \\
\alpha & \neg\alpha & \alpha & \alpha \\
\downarrow & \downarrow & \downarrow & \\
\beta & \neg\beta & \neg\beta &
\end{array}$$

Typ 2:

$$\begin{tikzcd}[column sep=small]
 & \neg(\alpha \wedge \beta) : & & & \alpha \vee \beta : & & & \alpha \rightarrow \beta : & \\
 & \bullet \arrow[dl] \arrow[dr] & & & \bullet \arrow[dl] \arrow[dr] & & & \bullet \arrow[dl] \arrow[dr] & \\
\neg\alpha & & \neg\beta & \alpha & & \beta & \neg\alpha & & \beta
\end{tikzcd}$$

Bsp: $((A \to B) \land (B \to C)) \to (A \to C))$ ist gültig

Wir zeigen $\models \alpha$, indem wir ein widersprüchliches Tableau für $\{\neg\alpha\}$ angeben:

$$\begin{tikzcd}[row sep=small]
 & \neg(((A \to B) \land (B \to C)) \to (A \to C)) \arrow[d] & & & \\
 & (A \to B) \land (B \to C) \arrow[d] & & & \\
 & \neg(A \to C) \arrow[d] & & & \\
 & A \to B \arrow[d] & & & \\
 & B \to C \arrow[d] & & & \\
 & A \arrow[d] & & & \\
 & \neg C \arrow[dl] \arrow[drr] & & & \\
\neg A & & & B \arrow[dl] \arrow[dr] & \\
\times & & \neg B & & C \\
 & & \times & & \times
\end{tikzcd}$$

3.3 Tableau-Beweisbarkeit

Wir definieren nun ein Beweissystem, das auf Tableaux basiert.

Da alle aussagenlogischen Formeln äquivalent zu Formeln aus Atomen, $\perp$ und $\rightarrow$ sind, reicht es dabei, Tableaux für solche Formeln zu definieren.

Ein Vorteil dieser Tableaux ist es,
dass sie mit zwei Expansionsregeln auskommen.

Das macht einen der Kern-Sätze dieses Kapitels, dass man mit dem Tableau-System genau die korrekten Folgerungen beweisen kann, einfacher beweisbar.

Definition 3.3 (Tableau für Formeln aus Atomen, $\perp$ und $\rightarrow$)

Sei Δ eine nicht-leere endliche Menge von Formeln aus Atomen, $\perp$ und $\rightarrow$. Wir fassen $\neg\alpha$ als abkürzende Schreibweise für $\alpha \rightarrow \perp$ auf.

1. Ein Pfad, dessen Knoten mit Formeln aus Δ markiert sind, ist ein Tableau für Δ.

2. Sei T ein Tableau für Δ,
und v sei ein Knoten in T mit Markierung ψ.
Wenn ψ eine Formel der Form $\alpha \rightarrow \beta$ (mit $\beta \neq \perp$) oder $\neg(\alpha \rightarrow \beta)$ ist, dann kann v mit der entsprechenden Expansionsregel expandiert werden.

$\alpha \rightarrow \beta$:
$$\begin{tikzcd} & \bullet \arrow[dl] \arrow[dr] & \\ \neg\alpha & & \beta \end{tikzcd}$$

$\neg(\alpha \rightarrow \beta)$:
$$\begin{tikzcd} \bullet \arrow[d] \\ \alpha \arrow[d] \\ \neg\beta \end{tikzcd}$$

Durch Expansion von v entsteht ein (weiteres) Tableau für Δ.

Bem.: Die Expansionsregel für $\neg\neg\alpha$ steckt in der für $\neg(\alpha \rightarrow \perp)$ ...

Lemma 3.4 (endliche Tableaux reichen)

Sei $\|\alpha\|$ *die Anzahl von Vorkommen von Atomen und* $\perp$ *in* α.

1. *Für jede Formel* α *gibt es ein geschlossenes Tableau mit* $\leqslant 2^{\|\alpha\|-1}$ *Pfaden.*
2. *Für jede endliche Formelmenge* $\Delta = \{\alpha_1, \ldots, \alpha_m\}$ *gibt es ein geschlossenes Tableau mit* $\leqslant 2^{\|\alpha_1\|+\ldots+\|\alpha_m\|}$ *Pfaden.*

Beispiel für $\|\alpha\|$:

$$\|(\perp \to (\perp \to \perp)) \to (A \to (\perp \to A))\| = 6$$

Jedes *Vorkommen* von Atomen oder $\perp$ wird mitgezählt.

$\|\alpha\|$ ist also (meistens) etwa die Hälfte von $|\alpha|$.

Beweis von 1. (der Beweis von 2. geht entsprechend):
Wir schreiben $\#\mathit{Pfade}(\alpha)$ für die Anzahl der Pfade im kleinsten geschlossenen Tableau für α.
Das Lemma sagt also aus: für alle Formeln α gilt $\#\mathit{Pfade}(\alpha) \leqslant 2^{\|\alpha\|-1}$.

Wir zeigen sogar etwas mehr:
für alle Formeln α gilt

(a) wenn $\alpha = \neg\beta$ für eine Formel β, dann gilt $\#\mathit{Pfade}(\alpha) \leqslant 2^{\|\alpha\|-2}$,
(b) und allgemein gilt $\#\mathit{Pfade}(\alpha) \leqslant 2^{\|\alpha\|-1}$

Wir führen eine Induktion über $\|\alpha\|$.

IA: α ist eine Formel mit $\|\alpha\| = 1$.
Dann ist α ein Atom oder $\perp$.
Also ist α eine Formel, die nicht expandiert wird.
Folglich ist $\#\mathit{Pfade}(\alpha) = 1 \leqslant 2^{\|\alpha\|-1}$.

IV: Für bel. festes *n* und alle Formeln α mit $\|\alpha\| \leqslant n$ gilt
$\#\mathit{Pfade}(\alpha) \leqslant 2^{\|\alpha\|-2}$ falls $\alpha = \neg\beta$, und $\#\mathit{Pfade}(\alpha) \leqslant 2^{\|\alpha\|-1}$.

IS: Sei φ eine Formel mit $\|\varphi\| = n + 1$.
Dann hat φ eine der drei folgenden Formen:

(1) $\varphi = \alpha \to \beta$ mit $\beta = \bot$ und α ist ein Atom oder $\bot$
(d.h. $\varphi = \neg A_i$ oder $\varphi = \neg\bot$)

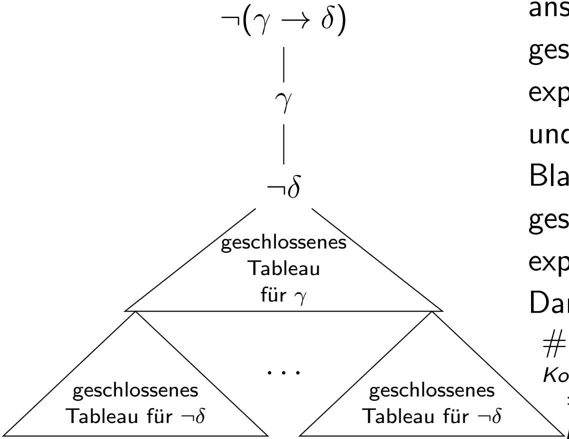
(2) $\varphi = \alpha \to \beta$ mit $\beta = \bot$ und $\alpha = \gamma \to \delta$
(d.h. $\varphi = \neg(\gamma \to \delta)$)

(3) $\varphi = \alpha \to \beta$ mit $\beta \neq \bot$

In Fall (1) und (2) ist $\#\mathit{Pfade}(\varphi) \leqslant 2^{\|\varphi\|-2}$ zu zeigen,
und in Fall (3) ist $\#\mathit{Pfade}(\varphi) \leqslant 2^{\|\varphi\|-1}$ zu zeigen.

Fall (1): $\varphi = \neg A_i$ oder $\varphi = \neg\bot$.
Dann wird φ nicht expandiert, es gilt $\|\varphi\| = 2$
und es folgt $\#\mathit{Pfade}(\varphi) = 1 \leqslant 2^{\|\varphi\|-2}$.

Fall (2): $\varphi = \neg(\gamma \to \delta)$.



Wir konstruieren ein geschlossenes Tableau für φ, indem wir zuerst φ expandieren, anschließend γ zu einem geschlossenen Tableau expandieren, und abschließend an jedem Blatt des Tableaus $\neg\delta$ zu einem geschlossenen Tableau expandieren.

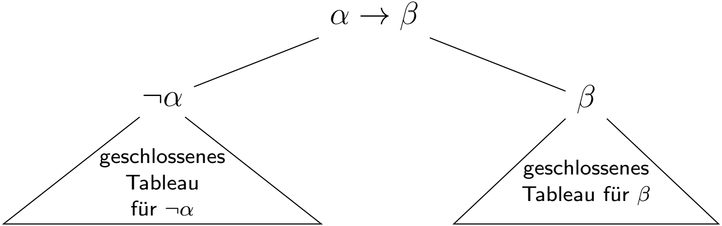
Dann gilt:

$$\begin{aligned}
&\#\mathit{Pfade}(\varphi) \\
&\overset{\mathit{Konstr.}}{\leqslant} \ \#\mathit{Pfade}(\gamma) \cdot \#\mathit{Pfade}(\neg\delta) \\
&\overset{\mathit{IV}}{\leqslant} \ 2^{\|\gamma\|-1} \cdot 2^{\|\neg\delta\|-2} \\
&\leqslant \ 2^{\|\gamma\|+\|\neg\delta\|-2} \\
&= \ 2^{\|\varphi\|-2}
\end{aligned}$$

Fall (3): $\varphi = \alpha \to \beta$ mit $\beta \neq \bot$

Wir konstruieren ein geschlossenes Tableau für φ, indem wir zuerst φ expandieren,

und anschließend $\neg\alpha$ und β zu geschlossenen Tableaux expandieren.



Dann gilt:

$$
\begin{array}{rll}
\#\mathit{Pfade}(\varphi) & & \\
\leqslant & \#\mathit{Pfade}(\neg\alpha) + \#\mathit{Pfade}(\beta) & \text{(Konstruktion des Tableaus)} \\
\leqslant & 2^{\|\neg\alpha\|-2} + 2^{\|\beta\|-1} & \text{(IV)} \\
\leqslant & 2^{\max\{\|\alpha\|-1, \|\beta\|-1\}+1} & \\
= & 2^{\max\{\|\alpha\|, \|\beta\|\}} & \\
\leqslant & 2^{\|\alpha\to\beta\|-1} & (\|\alpha\|, \|\beta\| \geqslant 1) \\
= & 2^{\|\varphi\|-1} & \checkmark
\end{array}
$$

Definition 3.5 (Tableau-beweisbar)

Sei Γ eine endliche Formelmenge und α eine Formel.
Die Relation $\vdash_{\overline{\mathsf{Tab}}}$ zwischen Formelmengen und Formeln ist definiert durch:
$\Gamma \vdash_{\overline{\mathsf{Tab}}} \alpha$ gdw. es ein widersprüchliches Tableau für $\Gamma \cup \{\neg\alpha\}$ gibt.

Man spricht „$\Gamma \vdash_{\overline{\mathsf{Tab}}} \alpha$“ aus als „$\alpha$ ist Tableau-beweisbar aus Γ“.

Wir werden zeigen (Satz (5.12)):

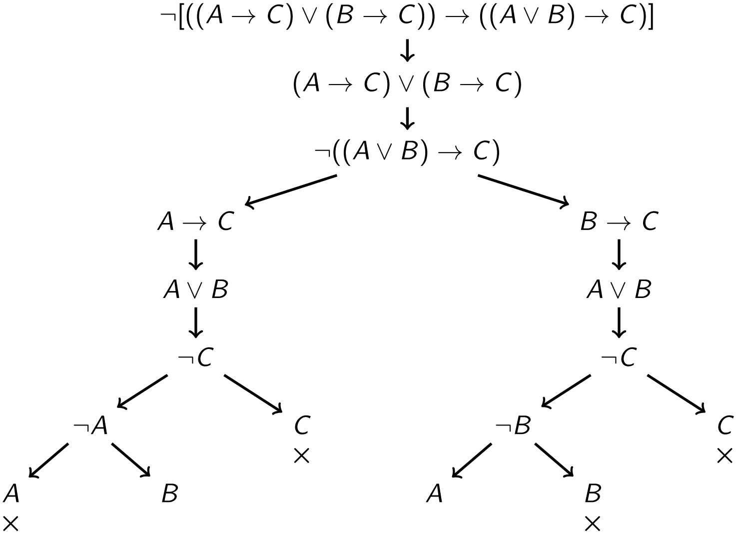
Für jede endliche Formelmenge Γ und jede Formel α gilt:
$\Gamma \vdash_{\overline{\mathsf{Tab}}} \alpha$ genau dann, wenn $\Gamma \Vdash \alpha$.

Das heißt also:

- $\models \alpha$ gdw. es ein widersprüchliches Tableau für $\{\neg\alpha\}$ gibt.
- $\Gamma \Vdash \alpha$ gdw. es ein widersprüchliches Tableau für $\Gamma \cup \{\neg\alpha\}$ gibt.

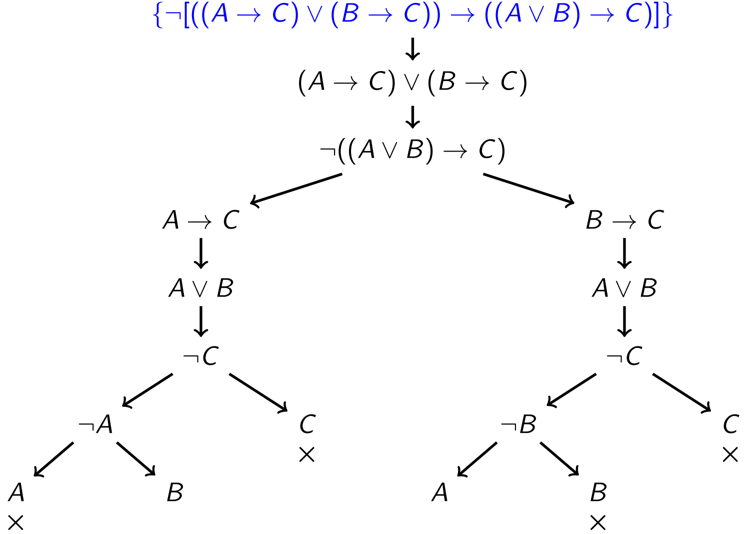
3.4 Algorithmische Umsetzung des Tableau-Aufbaus

Pfade werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



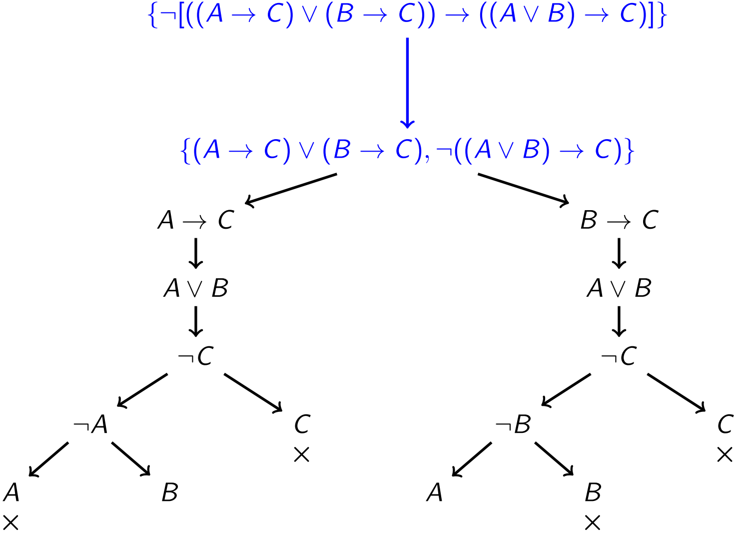
3.4 Algorithmische Umsetzung des Tableau-Aufbaus

Pfade werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



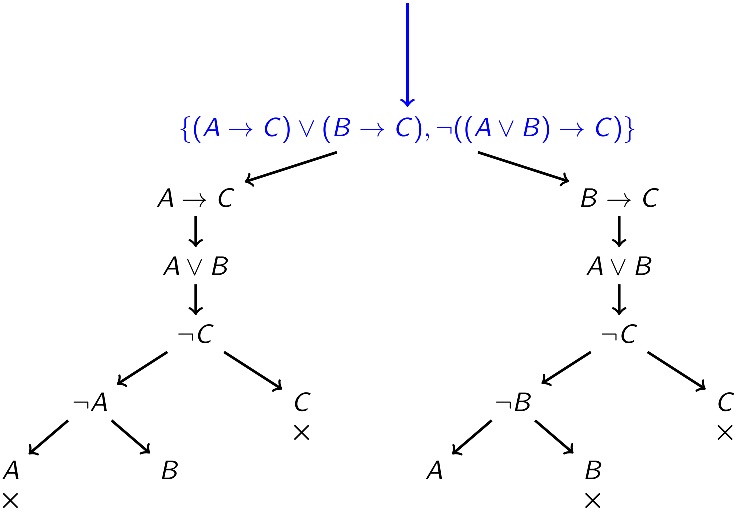
3.4 Algorithmische Umsetzung des Tableau-Aufbaus

Pfade werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



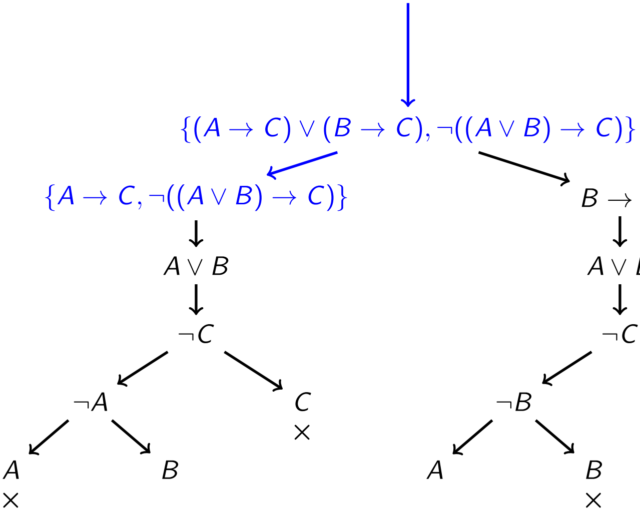
3.4 Algorithmische Umsetzung des Tableau-Aufbaus

Pfade werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



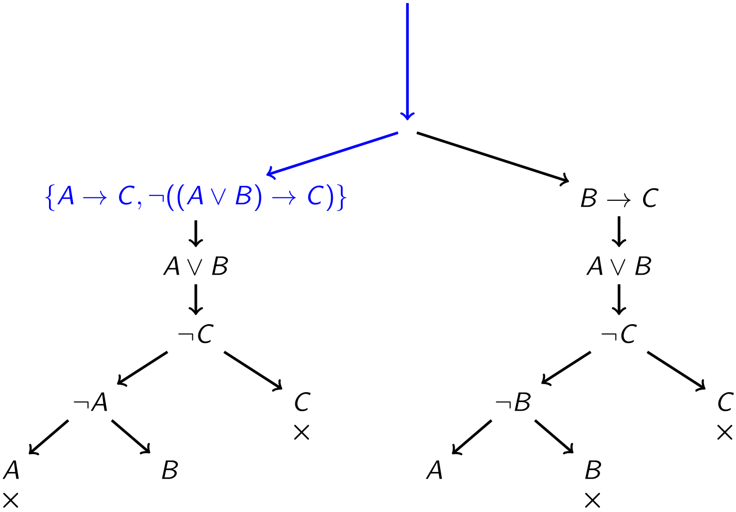
3.4 Algorithmische Umsetzung des Tableau-Aufbaus

Pfade werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



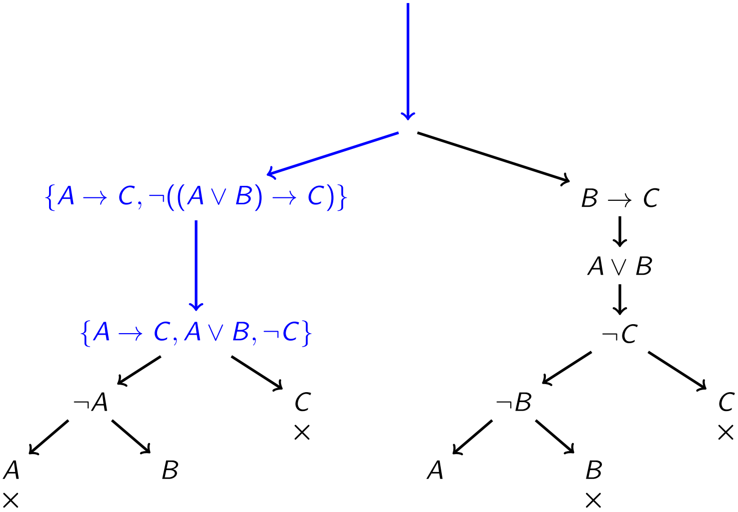
3.4 Algorithmische Umsetzung des Tableau-Aufbaus

Pfade werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



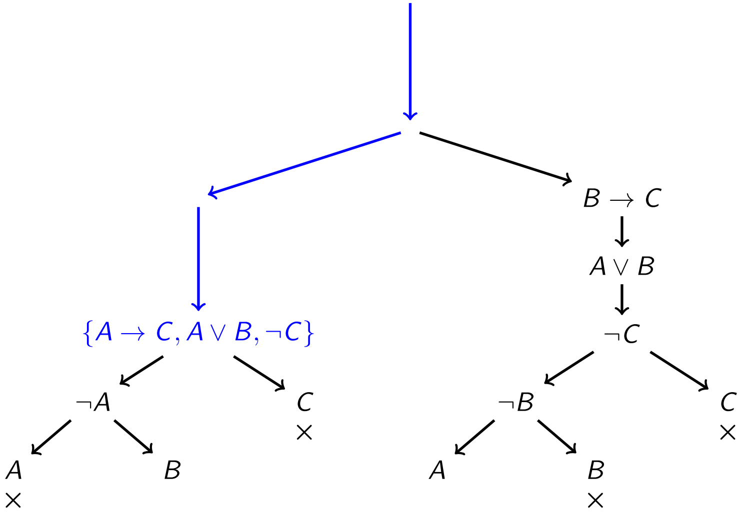
3.4 Algorithmische Umsetzung des Tableau-Aufbaus

Pfade werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



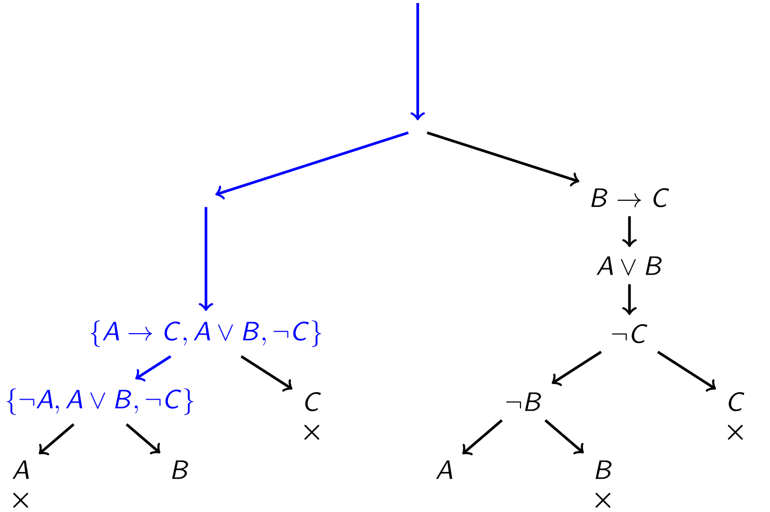
3.4 Algorithmische Umsetzung des Tableau-Aufbaus

Pfade werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



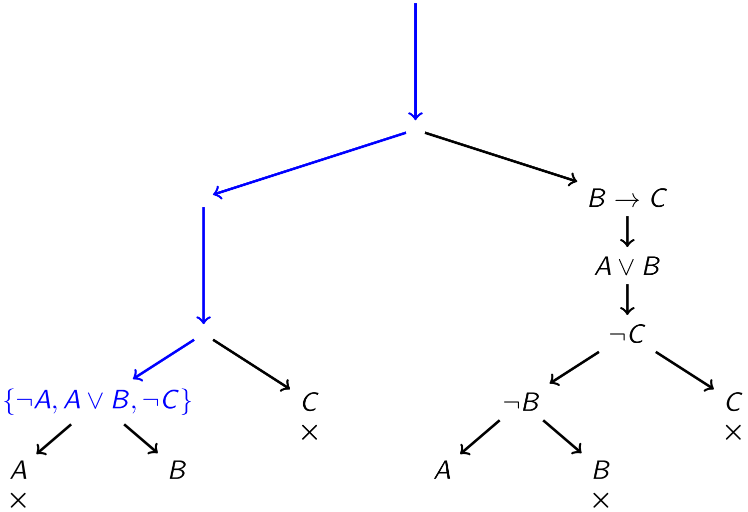
3.4 Algorithmische Umsetzung des Tableau-Aufbaus

Pfade werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



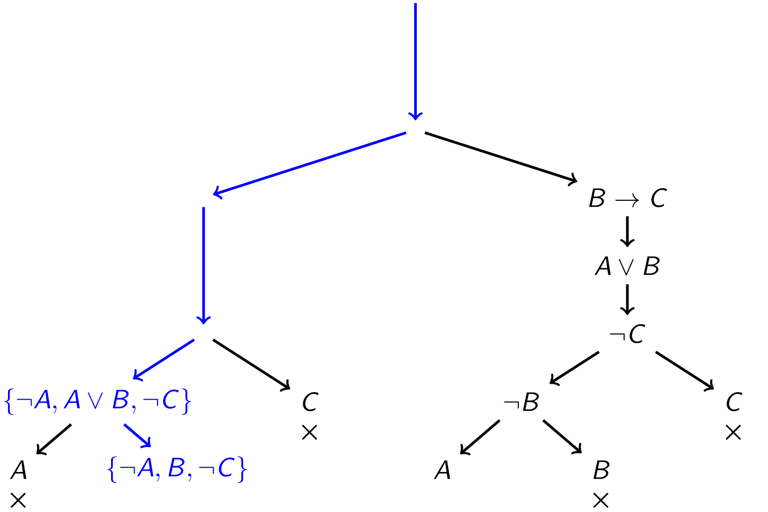
3.4 Algorithmische Umsetzung des Tableau-Aufbaus

Pfade werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



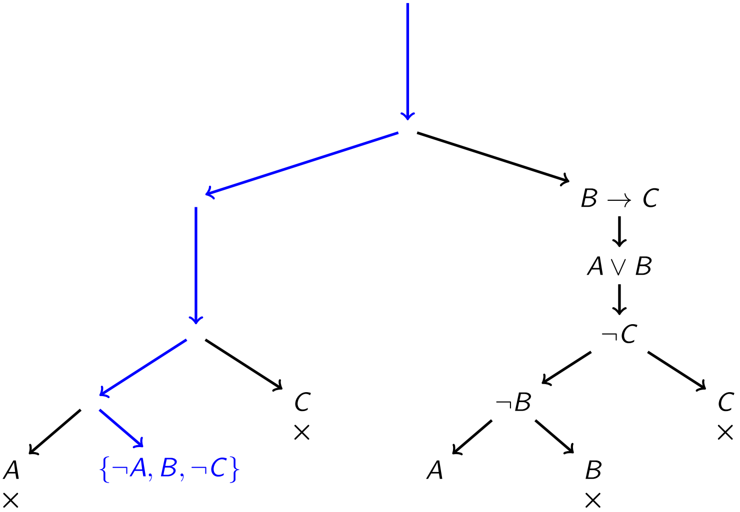
3.4 Algorithmische Umsetzung des Tableau-Aufbaus

Pfade werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



3.4 Algorithmische Umsetzung des Tableau-Aufbaus

Pfade werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



Jedes Tableau bestimmt einen Formelmengenbaum

$$
\begin{tikzcd}[column sep=tiny]
 & & & & \{\neg[((A \to C) \vee (B \to C)) \to ((A \vee B) \to C)]\} \arrow[d] & & & \\
 & & & & \{(A \to C) \vee (B \to C), \neg((A \vee B) \to C)\} \arrow[dll] \arrow[drr] & & & \\
 & & \{A \to C, \neg((A \vee B) \to C)\} \arrow[d] & & & & \{B \to C, \neg((A \vee B) \to C)\} \arrow[d] & \\
 & & \{A \to C, A \vee B, \neg C\} \arrow[dl] \arrow[dr] & & & & \{B \to C, A \vee B, \neg C\} \arrow[dl] \arrow[dr] & \\
 & \{\neg A, A \vee B, \neg C\} \arrow[dl] \arrow[dr] & & \{C, A \vee B, \neg C\} & & \{\neg B, A \vee B, \neg C\} \arrow[dl] \arrow[dr] & & \{C, A \vee B, \neg C\} \\
\{\neg A, A, \neg C\} & & \{\neg A, B, \neg C\} & \times & \{\neg B, A, \neg C\} & & \{\neg B, B, \neg C\} & \times \\
\times & & & & & & \times &
\end{tikzcd}
$$

Jede Menge repräsentiert einen Präfix eines Pfades durch das Tableau.

Jedes Tableau bestimmt einen Formelmengenbaum

- $\{\neg[((A \to C) \vee (B \to C)) \to ((A \vee B) \to C)]\}$
  - $\{(A \to C) \vee (B \to C), \neg((A \vee B) \to C)\}$
    - $\{A \to C, \neg((A \vee B) \to C)\}$
      - $\{A \to C, A \vee B, \neg C\}$
        - $\{\neg A, A \vee B, \neg C\}$
          - $\{\neg A, A, \neg C\}$ 0
          - $\{\neg A, B, \neg C\}$
        - $\{C, A \vee B, \neg C\}$ 0
    - $\{B \to C, \neg((A \vee B) \to C)\}$
      - $\{B \to C, A \vee B, \neg C\}$
        - $\{\neg B, A \vee B, \neg C\}$
          - $\{\neg B, A, \neg C\}$
          - $\{\neg B, B, \neg C\}$ 0
        - $\{C, A \vee B, \neg C\}$ 0

Jede Menge repräsentiert einen Präfix eines Pfades durch das Tableau.

Jedes Tableau bestimmt einen Formelmengenbaum

- $\{\neg[((A \to C) \lor (B \to C)) \to ((A \lor B) \to C)]\}$
  - $\{(A \to C) \lor (B \to C), \neg((A \lor B) \to C)\}$
    - $\{A \to C, \neg((A \lor B) \to C)\}$
      - $\{A \to C, A \lor B, \neg C\}$
        - $\{\neg A, A \lor B, \neg C\}$
          - $\{\neg A, A, \neg C\}$ 0
          - $\{\neg A, B, \neg C\}$ 1
        - $\{C, A \lor B, \neg C\}$ 0
    - $\{B \to C, \neg((A \lor B) \to C)\}$
      - $\{B \to C, A \lor B, \neg C\}$
        - $\{\neg B, A \lor B, \neg C\}$
          - $\{\neg B, A, \neg C\}$ 1
          - $\{\neg B, B, \neg C\}$ 0
        - $\{C, A \lor B, \neg C\}$ 0

Jede Menge repräsentiert einen Präfix eines Pfades durch das Tableau.

Jedes Tableau bestimmt einen Formelmengenbaum

$$
\begin{tikzcd}[column sep=tiny]
 & & & & & \{\neg[((A \to C) \lor (B \to C)) \to ((A \lor B) \to C)]\} \arrow[d] & & & & \\
 & & & & & \{(A \to C) \lor (B \to C), \neg((A \lor B) \to C)\} \arrow[dlll, "\text{max}"'] \arrow[drrr] & & & & \\
 & & \{A \to C, \neg((A \lor B) \to C)\} \arrow[d] & & & & & & \{B \to C, \neg((A \lor B) \to C)\} \arrow[d] & \\
 & & \{A \to C, A \lor B, \neg C\} \arrow[dl, "\text{max}"'] \arrow[dr] & & & & & & \{B \to C, A \lor B, \neg C\} \arrow[dl, "\text{max}"'] \arrow[dr] & \\
 & \{\neg A, A \lor B, \neg C\} \arrow[dl, "\text{max}"'] \arrow[dr] & & \{C, A \lor B, \neg C\} & & & & \{\neg B, A \lor B, \neg C\} \arrow[dl, "\text{max}"'] \arrow[dr] & & \{C, A \lor B, \neg C\} \\
\{\neg A, A, \neg C\} & & \{\neg A, B, \neg C\} & 0 & & & \{\neg B, A, \neg C\} & & \{\neg B, B, \neg C\} & 0 \\
0 & & 1 & & & & 1 & & 0 &
\end{tikzcd}
$$

Jede Menge repräsentiert einen Präfix eines Pfades durch das Tableau.

Idee für einen Algorithmus

Stelle mittels Tiefensuche durch den Formelmengenbaum fest,
dass alle Pfade des Tableaus widersprüchlich sind.

Das geht genauso wie:

Stelle mittels Tiefensuche durch den Formelmengenbaum fest,
dass es einen geschlossenen (nicht-widersprüchlichen) Pfad
durch das Tableau gibt.

Gültigkeitstest gemäß Tableau-Kalkül

für aussagenlogische Formeln mittels einer rekursiven Methode für die Tiefensuche durch den Formelmengenbaum

Methode erfüllbar(Formelmenge $\mathcal{S}$):

(∗ liefert Ergebnis 1, falls $\mathcal{S}$ erfüllbar ist, und Ergebnis 0 sonst ∗)

falls $\mathcal{S}$ widersprüchlich ist: return 0 (∗ $\mathcal{S}$ ist widersprüchlich ∗)

falls $\mathcal{S}$ eine Formel ψ der Form $\neg(\alpha \to \beta)$ enthält:

(∗ ersetze ψ durch alle ihre Zerlegungsformeln ∗)

return erfüllbar$((\mathcal{S} - \psi) \cup \{\alpha, \neg\beta\})$

sonst: falls $\mathcal{S}$ eine Formel ψ der Form $\alpha \to \beta$ mit $\beta \neq \bot$ enthält:

(∗ ersetze ψ „parallel" durch jeweils eine Zerlegungsformel ∗)

return $\max\limits_{\gamma \in \{\neg\alpha, \beta\}}$ erfüllbar$((\mathcal{S} - \psi) \cup \{\gamma\})$

sonst: (∗ $\mathcal{S}$ ist nicht widersprüchlich und enthält nur noch Literale ∗)

return 1 (∗ $\mathcal{S}$ ist erfüllende Belegung von φ ∗)

Tableau-Algorithmus (liefert Ergebnis 1, falls φ gültig ist, und Erg. 0 sonst)

Eingabe Formel φ

Ausgabe $1 -$ erfüllbar$(\{\neg\varphi\})$

Grobe Analyse des Tableau-Algorithmus

für Aussagenlogik

Der Algorithmus simuliert das Tableau-Verfahren.

Jeder Tiefensuchepfad entspricht einem Pfad durch das Tableau.

Die Menge $\mathcal{S}$ enthält stets die Knoten-Markierungen auf dem Pfad,
die noch nicht zerlegt/expandiert wurden.

Da jede Formel der Länge n höchstens $2n$ Teilformeln besitzt,
wird jeder Tiefensuchepfad in polynomieller Zeit durchlaufen,

Da das Tableau für eine Formel der Länge n höchsten 2^n Pfade hat,
hat der Algorithmus exponentielle Rechenzeit.

Das gleiche als nichtdeterministischer Algorithmus

nichtdeterministischer Tableau-Algorithmus:

Eingabe: Formelmenge $\mathcal{S}$

solange $\mathcal{S}$ nicht widersprüchlich und expandierbar ist, wiederhole:

falls $\mathcal{S}$ eine Formel ψ der Form $\neg(\alpha \to \beta)$ enthält:

(* ersetze ψ durch alle ihre Zerlegungsformeln *)

$\mathcal{S} = (\mathcal{S} - \psi) \cup \{\alpha, \neg\beta\}$

sonst: falls $\mathcal{S}$ eine Formel ψ der Form $\alpha \to \beta$ mit $\beta \neq \perp$ enthält:

(* wähle nichtdeterministisch eine Zerlegungsformel *)

wähle (existentiell) nichtdeterministisch $\gamma \in \{\neg\alpha, \beta\}$

$\mathcal{S} = (\mathcal{S} - \psi) \cup \{\gamma\}$

falls $\mathcal{S}$ nicht widersprüchlich ist: akzeptiere (* entspricht Ausgabe 1 *)

sonst: verwirf (* entspricht Ausgabe 0 *)

Es gilt:

(1) der nichtdetermistische Algorithmus hat bei Eingabe $\{\varphi\}$ einen akzeptierenden Berechnungspfad genau dann, wenn φ erfüllbar ist.

(2) der nichtdeterministische Algorithmus hat polynomielle Rechenzeit.

Was haben wir in Vorlesung 3 gelernt?

- Wir wissen, wie man mittels Expansionsregeln für verschiedene Verknüpfungszeichen ein Tableau für eine Formel oder eine Formelmenge aufbaut, und wie groß das Tableau höchstens werden kann, wenn man sich nicht zu ungeschickt anstellt.
- Wir kennen die Eigenschaften geschlossen und widersprüchlich von Tableaux.
- Wir kennen die Relation $\vdash_{\overline{\mathsf{Tab}}}$ der Tableau-Beweisbarkeit.
- Wir haben gehört, dass gültige Formeln genau die Tableau-beweisbaren Formeln sind.
- Wir wissen, wie man die Idee der Tableau-Konstruktion algorithmisch umsetzen kann. Dadurch erhält man einen determistischen Gültigkeitstest mit exponentieller Rechenzeit oder einen nichtdeterministischen Erfüllbarkeitstest mit polynomieller Rechenzeit („NP-Algorithmus“).

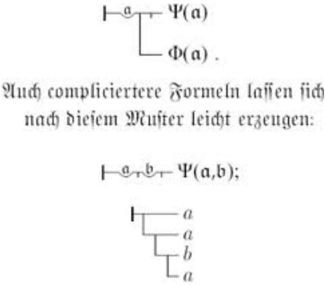
Vorlesung 4: Ein Frege-Kalkül

[Literatur: Mendelson: Introduction to Mathematical Logic]

Gottlob Frege (1848-1925)

Grundgesetze der Arithmetik (1893, 1902)

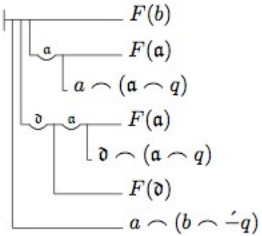
Gottlob Frege (1848-1925) war Professor in Jena.
Er wollte die Grundgesetze der Mathematik finden, aus denen sich jeder mathematische Satz herleiten lässt.
Dadurch wurde er zum Gründungsvater der formalen Logik.



Gottlob Frege (1848-1925)

Grundgesetze der Arithmetik (1893, 1902)

Gottlob Frege (1848-1925) war Professor in Jena.
Er wollte die Grundgesetze der Mathematik finden, aus denen sich jeder mathematische Satz herleiten lässt.
Dadurch wurde er zum Gründungsvater der formalen Logik.



Frege-Kalkülen liegt die Idee zu Grunde, dass man aus grundlegenden gültigen Formeln (Axiomen) alle anderen gültigen Formeln zusammensetzen kann. Die Regeln für das Zusammensetzen spiegeln das korrekte Ziehen von Schlüssen wider. Wir werden für den Frege-Kalkül nur Formeln aus Atomen, $\rightarrow$ und $\perp$ betrachten.
In dieser Vorlesung werden wir folgendes machen.

- Definition von Axiomen und Schlussregeln für Frege-Beweise.
- Werkzeuge zum Vereinfachen von Frege-Beweisen.
- (Viele) Beispiele für Frege-Beweise.

4.1 Axiome, Schlussregeln und Beweise

Definition 4.1 (Herleitung von Formeln im Frege-Kalkül)

Der Frege-Kalkül dient zur Herleitung von Formeln aus Axiomen.

1. Die *Elemente* des Frege-Kalküls sind
die aussagenlogischen Formeln aus Atomen, $\perp$ und $\rightarrow$
($\neg\alpha$ ist abkürzende Schreibweise für $\alpha \rightarrow \perp$; $\top$ ist Abkürzung für $\neg\perp$).

2. Die Axiome des Frege-Kalküls sind für alle Formeln α, β, φ:

(A1) $\alpha \rightarrow (\beta \rightarrow \alpha)$

(A2) $(\alpha \rightarrow (\beta \rightarrow \varphi)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \varphi))$

(A3) $(\neg\neg\alpha) \rightarrow \alpha$

3. Die einzige Schlussregel des Frege-Kalküls ist *modus ponens* (MP):
aus α und $\alpha \rightarrow \beta$ kann man in einem Schritt β herleiten.

Das wird auch beschrieben durch $\dfrac{\alpha \qquad \alpha\rightarrow\beta}{\beta}$.

(Fortsetzung von Definition 4.1)

4. Eine Herleitung einer Formel α aus einer Formelmenge Γ im Frege-Kalkül ist eine Folge $\alpha_1, \alpha_2, \ldots, \alpha_\ell$ von Formeln, an deren Ende $\alpha (= \alpha_\ell)$ steht und deren Elemente folgende Eigenschaften haben (für $i = 1, 2, \ldots, \ell$):

- α_i ist ein Axiom oder $\alpha_i \in \Gamma$ (α_i *ist eine Hypothese*), oder
- es gibt α_a, α_b mit $a, b < i$, aus denen α_i in einem Schritt mit modus ponens hergeleitet werden kann
  (d.h. es gibt $a, b < i$ mit $\alpha_b = \alpha_a \to \alpha_i$).

(Statt *Herleitung* verwendet man gerne auch (Frege-)Beweis.)

Bsp.: eine Herleitung für $B \to B$

$$\begin{aligned}
\alpha_1 &= B \to ((B \to B) \to B) && \text{Axiom (A1)} \\
\alpha_2 &= (B \to ((B \to B) \to B)) \to ((B \to (B \to B)) \to (B \to B)) && \\
& && \text{Axiom (A2)} \\
\alpha_3 &= (B \to (B \to B)) \to (B \to B) && \text{MP mit } \alpha_1 \text{ und } \alpha_2 \\
\alpha_4 &= B \to (B \to B) && \text{Axiom (A1)} \\
\alpha_5 &= B \to B && \text{MP mit } \alpha_4 \text{ und } \alpha_3
\end{aligned}$$

Nimmt man statt B eine beliebige Formel β,
dann hat man eine Herleitung für $\beta \to \beta$ für alle β.

Definition 4.2 (Frege-beweisbar)

Sei Γ eine Formelmenge und α eine Formel.
Die Relation $\vdash_{\mathsf{Fre}}$ zwischen Formelmengen und Formeln ist definiert durch:

$\Gamma \vdash_{\mathsf{Fre}} \alpha$ genau dann,
wenn es eine Herleitung von α aus Γ im Frege-Kalkül gibt.

„$\Gamma \vdash_{\mathsf{Fre}} \alpha$" spricht man aus als
„α ist Frege-herleitbar aus Γ" oder „α ist Frege-beweisbar aus Γ".

Für $\emptyset \vdash_{\mathsf{Fre}} \alpha$ schreibt man kurz $\vdash_{\mathsf{Fre}} \alpha$.
Eine Formel α mit $\vdash_{\mathsf{Fre}} \alpha$ nennt man auch *(Frege-)Theorem*.

Wir zeigen (wie im obigen Beispiel),
dass $\beta \to \beta$ ein Theorem des Frege-Kalküls ist.

Lemma 4.3

$\vdash_{\text{Fre}} \beta \to \beta$ *für jede Formel* β.

Beweis:

Sei β eine Formel.
Wir geben eine Herleitung von $\beta \to \beta$ im Frege-Kalkül an.

(1)	$\beta \to ((\beta \to \beta) \to \beta)$	(A1)
(2)	$(\beta \to ((\beta \to \beta) \to \beta)) \to ((\beta \to (\beta \to \beta)) \to (\beta \to \beta))$	(A2)
(3)	$(\beta \to (\beta \to \beta)) \to (\beta \to \beta)$	MP (1), (2)
(4)	$\beta \to (\beta \to \beta)$	(A1)
(5)	$\beta \to \beta$	MP (4), (3)

✓

Nun schauen wir uns Herleitungen mit Hypothesen an.

Lemma 4.4 (TRANS)

$\{\alpha \to \beta, \beta \to \gamma\} \vdash_{\mathsf{Fre}} \alpha \to \gamma$ *für alle Formeln* α, β, γ.

Beweis:
Seien α, β und γ Formeln.
Wir geben eine Herleitung von $\alpha \to \gamma$ aus $\alpha \to \beta$ und $\beta \to \gamma$ im Frege-Kalkül an.

(1)	$\beta \to \gamma$	Hypothese
(2)	$(\beta \to \gamma) \to (\alpha \to (\beta \to \gamma))$	(A1)
(3)	$\alpha \to (\beta \to \gamma)$	MP (1), (2)
(4)	$(\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma))$	(A2)
(5)	$(\alpha \to \beta) \to (\alpha \to \gamma)$	MP (3), (4)
(6)	$\alpha \to \beta$	Hypothese
(7)	$\alpha \to \gamma$	MP (6), (5)

✓

Wie bei $\models$ verzichten wir auch bei $\vdash_{\mathsf{Fre}}$ zur Abkürzung auf Mengenzeichen.

Lemma 4.5

$\bot \vdash_{\mathsf{Fre}} \alpha$ *für alle Formeln* α.

Beweis:
Sei α eine Formel.
Wir geben eine Herleitung von α aus $\bot$ im Frege-Kalkül an.

(1)	$\bot$	Hypothese
(2)	$\bot \to (\underbrace{\neg\alpha \to \bot}_{\neg\neg\alpha})$	(A1)
(3)	$\neg\neg\alpha$	MP (1), (2)
(4)	$\neg\neg\alpha \to \alpha$	(A3)
(5)	α	MP (3), (4) ✓

Wir wissen, dass $\bot \to \alpha$ gültig ist.
Es ist nicht soo leicht zu sehen, wie man $\bot \to \alpha$ herleiten kann.
Man kann aus der Herleitung von α aus $\bot$ eine Herleitung von $\bot \to \alpha$ (ohne Hypothese) machen, indem man systematisch $\bot \to$ vor jede hergeleitete Formel schreibt und ein paar Zwischenschritte einfügt ...

(1)	$\bot$	Hyp	$\bot \to \bot$	Lem. (4.3)
(2)	$\bot \to (\underbrace{\neg\alpha \to \bot}_{\neg\neg\alpha})$	(A1)	$\bot \to (\bot \to \neg\neg\alpha)$	MP(1.1),(1.2)
(3)	$\neg\neg\alpha$	MP(1),(2)	$\bot \to \neg\neg\alpha$	MP(1),(2.2)
(4)	$\neg\neg\alpha \to \alpha$	(A3)	$\bot \to (\neg\neg\alpha \to \alpha)$	MP(3.1),(3.2)
(5)	α	MP(3),(4)	$\bot \to \alpha$	MP (3),(4.2)

(1)	$\bot$	Hyp	$\bot \to \bot$	Lem. (4.3)
(1.1)			$\bot \to \neg\neg\alpha$	(A1)
(1.2)			$(\bot \to \neg\neg\alpha) \to (\bot \to (\bot \to \neg\neg\alpha))$	(A1)
(2)	$\bot \to (\underbrace{\neg\alpha \to \bot}_{\neg\neg\alpha})$	(A1)	$\bot \to (\bot \to \neg\neg\alpha)$	MP(1.1),(1.2)
(2.1)			$(\bot \to (\bot \to \neg\neg\alpha)) \to ((\bot \to \bot) \to (\bot \to \neg\neg\alpha))$	(A2)
(2.2)			$(\bot \to \bot) \to (\bot \to \neg\neg\alpha)$	MP(2),(2.1)
(3)	$\neg\neg\alpha$	MP(1),(2)	$\bot \to \neg\neg\alpha$	MP(1),(2.2)
(3.1)			$\neg\neg\alpha \to \alpha$	(A3)
(3.2)			$(\neg\neg\alpha \to \alpha) \to (\bot \to (\neg\neg\alpha \to \alpha))$	(A1)
(4)	$\neg\neg\alpha \to \alpha$	(A3)	$\bot \to (\neg\neg\alpha \to \alpha)$	MP(3.1),(3.2)
(4.1)			$(\bot \to (\neg\neg\alpha \to \alpha)) \to ((\bot \to \neg\neg\alpha) \to (\bot \to \alpha))$	(A2)
(4.2)			$(\bot \to \neg\neg\alpha) \to (\bot \to \alpha)$	MP(4),(4.1)
(5)	α	MP(3),(4)	$\bot \to \alpha$	MP (3),(4.2)

4.2 Werkzeuge zum Vereinfachen von Herleitungen

Satz 4.6 (Deduktionstheorem (DT))

Sei Γ *eine Menge von Formeln,* α *und* β *seien Formeln.*
Dann gilt: $\Gamma \cup \{\alpha\} \vdash_{\mathsf{Fre}} \beta$ *genau dann, wenn* $\Gamma \vdash_{\mathsf{Fre}} \alpha \to \beta$*.*

Beweis:

$\Leftarrow$: Sei $\Gamma \vdash_{\mathsf{Fre}} \alpha \to \beta$.
Dann gibt es folgenden Beweis mit Hypothesen aus $\Gamma \cup \{\alpha\}$.

$$
\begin{array}{lll}
(1) & \dots & \\
\vdots & \vdots & \left.\right\} \text{hier steht der Beweis für } \Gamma \vdash_{\mathsf{Fre}} \alpha \to \beta \\
(k) & \alpha \to \beta & \\
(k+1) & \alpha & \text{Hypothese} \\
(k+2) & \beta & \text{MP } (k+1), (k)
\end{array}
$$

Damit haben wir einen Beweis von β aufgeschrieben, in dem Hypothesen aus Γ (Schritte (1)–(k)) und Hypothese α benutzt wird (Schritt $(k+1)$).
Also haben wir $\Gamma \cup \{\alpha\} \vdash_{\mathsf{Fre}} \beta$.

$\Rightarrow$: Wir führen einen Induktionsbeweis über die Länge ℓ
von Herleitungen $\alpha_1, \ldots, \alpha_\ell$ von β aus $\Gamma \cup \{\alpha\}$.

IA $\ell = 1$: Sei $\Gamma \cup \{\alpha\} \vdash_{\text{Fre}} \beta$ mit einer Herleitung der Länge 1.
Dann ist β ein Axiom oder $\beta \in \Gamma \cup \{\alpha\}$

(andere Herleitungen von β mit Länge 1 gibt es nicht).

Zu zeigen ist nun: $\Gamma \vdash_{\text{Fre}} \alpha \to \beta$.

Fall 1: β ist Axiom oder $\beta \in \Gamma$:
dann können wir folgende Herleitung aus Γ aufschreiben:

(1)	β	Hyp. aus Γ oder Axiom
(2)	$\beta \to (\alpha \to \beta)$	(A1)
(3)	$\alpha \to \beta$	MP (1),(2)

Also folgt $\Gamma \vdash_{\text{Fre}} \alpha \to \beta$.

Fall 2: $\beta = \alpha$: es gilt $\vdash_{\text{Fre}} \beta \to \beta$ für jede Formel β (4.3).
Also gilt auch $\Gamma \vdash_{\text{Fre}} \beta \to \beta$, d.h. hier $\Gamma \vdash_{\text{Fre}} \alpha \to \beta$.

Das sind alle Möglichkeiten für β, und stets folgt $\Gamma \vdash_{\text{Fre}} \alpha \to \beta$.

IV: Für bel. festes k und für alle Γ, α, β gilt:

Wenn $\Gamma \cup \{\alpha\} \vdash_{\mathsf{Fre}} \beta$ mittels einer Herleitung mit $\ell \leq k$ Schritten,

dann folgt $\Gamma \vdash_{\mathsf{Fre}} \alpha \to \beta$.

IS: z.z: Wenn $\Gamma \cup \{\alpha\} \vdash_{\mathsf{Fre}} \beta$ mittels einer Herleitung aus $k+1$ Schritten,

dann folgt $\Gamma \vdash_{\mathsf{Fre}} \alpha \to \beta$.

Sei $\alpha_1, \ldots, \alpha_{k+1}$ eine Herleitung von β $(= \alpha_{k+1})$ aus $\Gamma \cup \{\alpha\}$.

Fall 1: β ist Axiom oder $\beta \in \Gamma \cup \{\alpha\}$.
Dann folgt $\Gamma \vdash_{\mathsf{Fre}} \alpha \to \beta$ wie im IA.

Fall 2: β entsteht mit MP aus α_i und α_j $(i, j \leq k)$.

Dann ist $\alpha_j = \alpha_i \to \beta$.

Nach IV gilt $\Gamma \vdash_{\text{Fre}} \alpha \to (\alpha_i \to \beta)$ und $\Gamma \vdash_{\text{Fre}} \alpha \to \alpha_i$.
Folgende Herleitung zeigt $\Gamma \vdash_{\text{Fre}} \alpha \to \beta$.

$$
\begin{array}{lll}
(1) & \dots & \\
\vdots & \vdots & \Big\} \text{ Herleitung von } \alpha \to (\alpha_i \to \beta) \text{ aus } \Gamma \text{ gemäß IV} \\
(s) & \alpha \to (\alpha_i \to \beta) & \\
(s+1) & \dots & \\
\vdots & \vdots & \Big\} \text{ Herleitung von } \alpha \to \alpha_i \text{ aus } \Gamma \text{ gemäß IV} \\
(m) & \alpha \to \alpha_i & \\
(m+1) & (\alpha \to (\alpha_i \to \beta)) \to ((\alpha \to \alpha_i) \to (\alpha \to \beta)) & \text{(A2)} \\
(m+2) & (\alpha \to \alpha_i) \to (\alpha \to \beta) & \text{MP } (s), (m+1) \\
(m+3) & \alpha \to \beta & \text{MP } (m), (m+2)
\end{array}
$$

Damit haben wir eine Herleitung von $\alpha \to \beta$ aus Hypothesen in Γ,
d.h. $\Gamma \vdash_{\text{Fre}} \alpha \to \beta$. ✓

Beispiele: Anwendung des Deduktionstheorems

zur Vereinfachung von Herleitungen

Lemma 4.7 (Die fehlende Richtung des Doppelnegationsgesetzes)

$\vdash_{\mathsf{Fre}} \alpha \to \neg\neg\alpha$ *für alle Formeln* α.

Beweis:

Zuerst zeigen wir $\alpha, \neg\alpha \vdash_{\mathsf{Fre}} \perp$ mit folgender Herleitung.

(1) α Hyp
(2) $\neg\alpha$ Hyp
(3) $\perp$ MP (1),(2) (Nun ist $\alpha, \neg\alpha \vdash_{\mathsf{Fre}} \perp$ bewiesen.)

Aus $\alpha, \neg\alpha \vdash_{\mathsf{Fre}} \perp$ folgt

$$\alpha \vdash_{\mathsf{Fre}} \underbrace{(\neg\alpha) \to \perp}_{=\neg\neg\alpha}$$ mit dem Deduktionstheorems (4.6).

Aus $\alpha \vdash_{\mathsf{Fre}} \neg\neg\alpha$ folgt

$\vdash_{\mathsf{Fre}} \alpha \to \neg\neg\alpha$ mit dem Deduktionstheorems (4.6). ✓

Um $\vdash_{\text{Fre}} \alpha \to \neg\neg\alpha$ zu zeigen,
haben wir jetzt nicht mehr die ganze Herleitung von $\alpha \to \neg\neg\alpha$ angegeben,
sondern nur noch argumentiert,
dass man aus einer Herleitung von $\bot$ aus den Hypothesen $\{\alpha, \neg\alpha\}$
mit dem Deduktionstheorem die Existenz einer Herleitung von $\alpha \to \neg\neg\alpha$
folgern kann.

Deshalb verallgemeinern wir jetzt die Schreibweise für Herleitungen
zu einer Schreibweise für die Existenz von Herleitungen.
Darin können wir Herleitungsschritte wie bisher im Frege-Kalkül begründen
und verallgemeinerte Herleitungsschritte durch das Deduktionstheorem oder
andere Lemmas begründen.

Beispiel für das Aufschreiben des letzten Beweises
als verallgemeinerte Herleitung (Mengenzeichen werden weggelassen):

(1)	$\alpha, \neg\alpha$	$\vdash_{\text{Fre}}$	α	Hyp	
(2)	$\alpha, \neg\alpha$	$\vdash_{\text{Fre}}$	$\neg\alpha$	Hyp	
(3)	$\alpha, \neg\alpha$	$\vdash_{\text{Fre}}$	$\bot$	MP (1),(2)	
(4)	α	$\vdash_{\text{Fre}}$	$\neg\neg\alpha$	DT (3)	$(\neg\alpha \to \bot = \neg\neg\alpha)$
(5)		$\vdash_{\text{Fre}}$	$\alpha \to \neg\neg\alpha$	DT (4)	

Lemma 4.8 (ex falso quod libet)

$\vdash_{\text{Fre}} \bot \to \alpha$ *für alle Formeln α.*

Beweis (anders als im Beispiel vor dem Deduktionstheorem):

Zuerst zeigen wir $\neg\alpha, \bot \vdash_{\text{Fre}} \bot$ (die Herleitung hat nur einen Schritt):

(1) $\bot$ Hyp

Aus $\neg\alpha, \bot \vdash_{\text{Fre}} \bot$ folgt $\bot \vdash_{\text{Fre}} \neg\neg\alpha$ mit dem Deduktionstheorem (4.6).

Nun nehmen wir eine Herleitung von $\neg\neg\alpha$ aus $\{\bot\}$ und hängen noch etwas daran:

(1)	$\cdots$	
$\vdots$	$\vdots$	(Herleitung von $\neg\neg\alpha$ aus $\{\bot\}$)
(m)	$\neg\neg\alpha$	
($m+1$)	$\neg\neg\alpha \to \alpha$	(A3)
($m+2$)	α	MP (m),($m+1$)

Das ist eine Herleitung von α aus $\{\bot\}$.
Damit ist $\bot \vdash_{\text{Fre}} \alpha$ bewiesen.
Daraus folgt mit dem Deduktionstheorem (4.6) $\vdash_{\text{Fre}} \bot \to \alpha$.

Der Beweis als verallgemeinerte Herleitung:

$$
\begin{array}{llcll}
(1) & \neg\alpha, \bot & \vdash_{\mathsf{Fre}} & \bot & \text{Hyp} \\
(2) & \bot & \vdash_{\mathsf{Fre}} & \neg\neg\alpha & \text{DT (1)} \\
(3) & & \vdash_{\mathsf{Fre}} & \neg\neg\alpha \to \alpha & \text{(A3)} \\
(4) & \bot & \vdash_{\mathsf{Fre}} & \alpha & \text{MP (2),(3)} \quad \text{(Hypothesenmengen vereinigen!)} \\
(5) & & \vdash_{\mathsf{Fre}} & \bot \to \alpha & \text{DT (4)} \qquad \checkmark
\end{array}
$$

Wenn man eine verallgemeinerte Herleitung hat,
dann weiß man, dass (und wie) man daraus eine Herleitung im Frege-Kalkül „basteln“ kann (und spart sich den mühseligen technischen Aufwand dazu).

Lemma 4.9

Sei Γ eine Formelmenge, und α sei eine Formel.
Aus $\Gamma, \neg\alpha \vdash_{\mathsf{Fre}} \bot$ folgt $\Gamma \vdash_{\mathsf{Fre}} \alpha$.

Beweis (als verallgemeinerte Herleitung):

(1)	$\Gamma, \neg\alpha$	$\vdash_{\mathsf{Fre}}$	$\bot$	Voraussetzung
(2)	Γ	$\vdash_{\mathsf{Fre}}$	$\neg\neg\alpha$	DT (1)
(3)		$\vdash_{\mathsf{Fre}}$	$\neg\neg\alpha \rightarrow \alpha$	(A3)
(4)	Γ	$\vdash_{\mathsf{Fre}}$	α	MP (2),(3) (Hypothesenmengen vereinigen!)

Formeln, deren Herleitung bereits bekannt ist (Theoreme),
können in verallgemeinerte Herleitungen eingebaut werden.

Lemma 4.10 (ex falso quod libet (allgemeiner))

$\vdash_{\mathsf{Fre}} \neg\alpha \to (\alpha \to \beta)$ *für alle Formeln α und β.*

Beweis:

(1)	α	$\vdash_{\mathsf{Fre}}$	α	Hyp
(2)	$\neg\alpha$	$\vdash_{\mathsf{Fre}}$	$\neg\alpha$	Hyp
(3)	$\alpha, \neg\alpha$	$\vdash_{\mathsf{Fre}}$	$\bot$	MP (1),(2)
(4)		$\vdash_{\mathsf{Fre}}$	$\bot \to \beta$	(4.8)
(5)	$\alpha, \neg\alpha$	$\vdash_{\mathsf{Fre}}$	β	MP (3),(4)
(6)	$\neg\alpha$	$\vdash_{\mathsf{Fre}}$	$\alpha \to \beta$	DT (5)
(7)		$\vdash_{\mathsf{Fre}}$	$\neg\alpha \to (\alpha \to \beta)$	DT (6)

✓

Lemma 4.11 (Herleitungen ersetzen Hypothesen)

Seien Δ und Γ Formelmengen, α und β seien Formeln.
Aus $\Gamma, \alpha \vdash_{\overline{\text{Fre}}} \beta$ und $\Delta \vdash_{\overline{\text{Fre}}} \alpha$ folgt $\Gamma, \Delta \vdash_{\overline{\text{Fre}}} \beta$.

Beweis:
Aus $\Gamma, \alpha \vdash_{\overline{\text{Fre}}} \beta$ folgt $\Gamma \vdash_{\overline{\text{Fre}}} \alpha \to \beta$ mit dem Deduktionstheorem (4.6).
Aus $\Delta \vdash_{\overline{\text{Fre}}} \alpha$ folgt $\Gamma, \Delta \vdash_{\overline{\text{Fre}}} \alpha$, da eine Herleitung mit Hypothesen aus Δ auch eine Herleitung mit Hypothesen aus $\Gamma \cup \Delta$ ist.
Entsprechend folgt $\Gamma, \Delta \vdash_{\overline{\text{Fre}}} \alpha \to \beta$ aus $\Gamma \vdash_{\overline{\text{Fre}}} \alpha \to \beta$.
Aus der Herleitung $\varphi_1, \ldots, \varphi_m$ von α aus $\Gamma \cup \Delta$
und der Herleitung $\psi_1, \ldots, \psi_\ell$ von $\alpha \to \beta$ aus $\Gamma \cup \Delta$
erhält man eine Herleitung $\varphi_1, \ldots, \varphi_m, \psi_1, \ldots, \psi_\ell, \beta$ aus $\Gamma \cup \Delta$, bei der β durch MP aus $\varphi_m = \alpha$ und $\psi_\ell = \alpha \to \beta$ entsteht.
Also gilt $\Gamma, \Delta \vdash_{\overline{\text{Fre}}} \beta$. ✓

Lemma 4.12 (Verallgemeinerung von TRANS)

Für jedes $k \geqslant 1$ und alle Formeln $\alpha_1, \ldots, \alpha_k, \beta, \gamma$ gilt
$\alpha_1 \to (\alpha_2 \to (\ldots \to (\alpha_k \to \beta) \ldots)), \beta \to \gamma \vdash_{\mathsf{Fre}} \alpha_1 \to (\alpha_2 \to (\ldots \to (\alpha_k \to \gamma) \ldots))$.

Beweis mittels Induktion über k.

IA $k = 1$: für $k = 1$ ist die Behauptung gleich TRANS (4.4).

IV: die Behauptung gilt für bel. festes k.

[Also gilt z.B. $\alpha_2 \to (\ldots \to (\alpha_{k+1} \to \beta) \ldots), \beta \to \gamma \vdash_{\mathsf{Fre}} \alpha_2 \to (\ldots \to (\alpha_{k+1} \to \gamma) \ldots)$.]

IS: zu zeigen: $\overbrace{\alpha_1 \to (\alpha_2 \to (\ldots \to (\alpha_{k+1} \to \beta) \ldots))}^{=:\psi}, \beta \to \gamma \vdash_{\mathsf{Fre}}$
$\alpha_1 \to (\alpha_2 \to (\ldots \to (\alpha_{k+1} \to \gamma) \ldots))$

(1)	$\psi, \beta \to \gamma$	$\vdash_{\mathsf{Fre}}$	$\alpha_1 \to (\alpha_2 \to (\ldots \to (\alpha_{k+1} \to \beta) \ldots))$	Hyp
(2)	$\psi, \beta \to \gamma, \alpha_1$	$\vdash_{\mathsf{Fre}}$	$\alpha_2 \to (\ldots \to (\alpha_{k+1} \to \beta) \ldots)$	DT (1)
(3)	$\psi, \beta \to \gamma, \alpha_1$	$\vdash_{\mathsf{Fre}}$	$\alpha_2 \to (\ldots \to (\alpha_{k+1} \to \gamma) \ldots)$	IV & (4.11) mit (2)
(4)	$\psi, \beta \to \gamma$	$\vdash_{\mathsf{Fre}}$	$\alpha_1 \to (\alpha_2 \to (\ldots \to (\alpha_{k+1} \to \gamma) \ldots))$	DT (3) ✓

Satz 4.13 („Wichtige“ Theoreme des Frege-Kalküls)

Für alle Formeln α und β gilt:

1. $\vdash_{\mathsf{Fre}} \alpha \to \alpha$ *(4.3)*
2. $\vdash_{\mathsf{Fre}} \alpha \to \neg\neg\alpha$ *(4.7)*
3. $\vdash_{\mathsf{Fre}} \neg\alpha \to (\alpha \to \beta)$ *(4.10)*
4. $\vdash_{\mathsf{Fre}} (\alpha \to \beta) \to (\neg\beta \to \neg\alpha)$
5. $\vdash_{\mathsf{Fre}} \alpha \to (\neg\beta \to \neg(\alpha \to \beta))$
6. $\vdash_{\mathsf{Fre}} (\alpha \to \beta) \to ((\neg\alpha \to \beta) \to \beta)$
7. $\vdash_{\mathsf{Fre}} \neg(\alpha \to \beta) \to \alpha$
8. $\vdash_{\mathsf{Fre}} \neg(\alpha \to \beta) \to \neg\beta$

(4) $\vdash_{\mathsf{Fre}} (\alpha \to \beta) \to (\neg\beta \to \neg\alpha)$

(1)	$\alpha \to \beta, \neg\beta, \alpha$	$\vdash_{\mathsf{Fre}}$	$\alpha \to \beta$	Hyp
(2)	$\alpha \to \beta, \neg\beta, \alpha$	$\vdash_{\mathsf{Fre}}$	$\neg\beta$	Hyp
(3)	$\alpha \to \beta, \neg\beta, \alpha$	$\vdash_{\mathsf{Fre}}$	α	Hyp
(4)	$\alpha \to \beta, \neg\beta, \alpha$	$\vdash_{\mathsf{Fre}}$	β	MP (1),(3)
(5)	$\alpha \to \beta, \neg\beta, \alpha$	$\vdash_{\mathsf{Fre}}$	$\perp$	MP (2),(4)
(6)	$\alpha \to \beta, \neg\beta$	$\vdash_{\mathsf{Fre}}$	$\neg\alpha$	DT (5)
(7)	$\alpha \to \beta$	$\vdash_{\mathsf{Fre}}$	$\neg\beta \to \neg\alpha$	DT (6)
(8)		$\vdash_{\mathsf{Fre}}$	$(\alpha \to \beta) \to (\neg\beta \to \neg\alpha)$	DT (7)

(5) $\vdash_{\text{Fre}} \alpha \to (\neg\beta \to \neg(\alpha \to \beta))$

(1)	$\alpha, \neg\beta, \alpha \to \beta$	$\vdash_{\text{Fre}}$	$\alpha \to \beta$	Hyp
(2)	$\alpha, \neg\beta, \alpha \to \beta$	$\vdash_{\text{Fre}}$	$\neg\beta$	Hyp
(3)	$\alpha, \neg\beta, \alpha \to \beta$	$\vdash_{\text{Fre}}$	α	Hyp
(4)	$\alpha, \neg\beta, \alpha \to \beta$	$\vdash_{\text{Fre}}$	β	MP (1),(3)
(5)	$\alpha, \neg\beta, \alpha \to \beta$	$\vdash_{\text{Fre}}$	$\perp$	MP (2),(4)
(6)	$\alpha, \neg\beta$	$\vdash_{\text{Fre}}$	$\neg(\alpha \to \beta)$	DT (5)
(7)	α	$\vdash_{\text{Fre}}$	$\neg\beta \to \neg(\alpha \to \beta)$	DT (6)
(8)		$\vdash_{\text{Fre}}$	$\alpha \to (\neg\beta \to \neg(\alpha \to \beta))$	DT (7)

(6) $\vdash_{\text{Fre}} (\alpha \to \beta) \to ((\neg\alpha \to \beta) \to \beta)$

(1)	$\alpha \to \beta, \neg\alpha \to \beta, \neg\beta$	$\vdash_{\text{Fre}}$	$\alpha \to \beta$	Hyp
(2)	$\alpha \to \beta, \neg\alpha \to \beta, \neg\beta$	$\vdash_{\text{Fre}}$	$\neg\alpha \to \beta$	Hyp
(3)	$\alpha \to \beta, \neg\alpha \to \beta, \neg\beta$	$\vdash_{\text{Fre}}$	$(\alpha \to \beta) \to (\neg\beta \to \neg\alpha)$	Satz 4.13(4)
(4)	$\alpha \to \beta, \neg\alpha \to \beta, \neg\beta$	$\vdash_{\text{Fre}}$	$\neg\beta \to \neg\alpha$	MP (1),(3)
(5)	$\alpha \to \beta, \neg\alpha \to \beta, \neg\beta$	$\vdash_{\text{Fre}}$	$(\neg\alpha \to \beta) \to (\neg\beta \to \neg\neg\alpha)$	Satz 4.13(4)
(6)	$\alpha \to \beta, \neg\alpha \to \beta, \neg\beta$	$\vdash_{\text{Fre}}$	$\neg\beta \to \neg\neg\alpha$	MP (2),(5)
(7)	$\alpha \to \beta, \neg\alpha \to \beta, \neg\beta$	$\vdash_{\text{Fre}}$	$\neg\beta$	Hyp
(8)	$\alpha \to \beta, \neg\alpha \to \beta, \neg\beta$	$\vdash_{\text{Fre}}$	$\neg\alpha$	MP (7),(4)
(9)	$\alpha \to \beta, \neg\alpha \to \beta, \neg\beta$	$\vdash_{\text{Fre}}$	$\neg\neg\alpha$	MP (7),(6)
(10)	$\alpha \to \beta, \neg\alpha \to \beta, \neg\beta$	$\vdash_{\text{Fre}}$	$\neg\neg\alpha \to \alpha$	(A3)
(11)	$\alpha \to \beta, \neg\alpha \to \beta, \neg\beta$	$\vdash_{\text{Fre}}$	α	MP (9),(10)
(12)	$\alpha \to \beta, \neg\alpha \to \beta, \neg\beta$	$\vdash_{\text{Fre}}$	$\bot$	MP (8),(11)
(13)	$\alpha \to \beta, \neg\alpha \to \beta$	$\vdash_{\text{Fre}}$	β	(4.9) (12)
(14)	$\alpha \to \beta$	$\vdash_{\text{Fre}}$	$(\neg\alpha \to \beta) \to \beta$	DT (13)
(15)		$\vdash_{\text{Fre}}$	$(\alpha \to \beta) \to ((\neg\alpha \to \beta) \to \beta)$	DT (14)

(7) $\vdash_{\mathsf{Fre}} \neg(\alpha \to \beta) \to \alpha$

(1)	$\vdash_{\mathsf{Fre}}$	$\neg\alpha \to (\alpha \to \beta)$	Satz 4.13(3)
(2)	$\vdash_{\mathsf{Fre}}$	$(\neg\alpha \to (\alpha \to \beta)) \to (\neg(\alpha \to \beta) \to \neg\neg\alpha)$	Satz 4.13(4)
(3)	$\vdash_{\mathsf{Fre}}$	$\neg(\alpha \to \beta) \to \neg\neg\alpha$	MP (1),(2)
(4)	$\vdash_{\mathsf{Fre}}$	$\neg\neg\alpha \to \alpha$	(A3)
(5)	$\vdash_{\mathsf{Fre}}$	$\neg(\alpha \to \beta) \to \alpha$	TRANS & (4.11) mit (3),(4)

(8) $\vdash_{\mathsf{Fre}} \neg(\alpha \to \beta) \to \neg\beta$

(1)	$\vdash_{\mathsf{Fre}}$	$\beta \to (\alpha \to \beta)$	(A1)
(2)	$\vdash_{\mathsf{Fre}}$	$(\beta \to (\alpha \to \beta)) \to (\neg(\alpha \to \beta) \to \neg\beta)$	Satz 4.13(4)
(3)	$\vdash_{\mathsf{Fre}}$	$\neg(\alpha \to \beta) \to \neg\beta$	MP (1),(2)

4.3 Ähnliche Kalküle

Die Auswahl der Axiome in „unserem" Frege-Kalkül hat das Ziel, den Beweis des Vollständigkeitssatzes technisch einfach zu machen.
Eine Theorie von Kleene (1952) hat Modus Ponens und die folgenden Axiome (für Formeln mit Verknüpfungszeichen $\to$, $\wedge$, $\vee$ und $\neg$).

1. $\alpha \to (\beta \to \alpha)$
2. $(\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma))$
3. $(\alpha \wedge \beta) \to \alpha$ und $(\alpha \wedge \beta) \to \beta$
4. $\alpha \to (\beta \to (\alpha \wedge \beta))$
5. $\alpha \to (\alpha \vee \beta)$ und $\beta \to (\alpha \vee \beta)$
6. $(\alpha \to \gamma) \to ((\beta \to \gamma) \to ((\alpha \vee \beta) \to \gamma))$
7. $(\alpha \to \beta) \to ((\alpha \to \neg\beta) \to \neg\alpha)$
8. $\neg\neg\alpha \to \alpha$

Durch Weglassen des letzten Axioms (Doppelnegationsgesetz) erhält man einen Kalkül für die *intuitionistische Logik*.

Ein Kalkül, der im Buch von Mendelson benutzt wird, hat Modus Ponens und die folgenden Axiome (für Formeln mit Verknüpfungszeichen $\to$ und $\neg$).

1. $\alpha \to (\beta \to \alpha)$

2. $(\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma))$

3. $(\neg\beta \to \neg\alpha) \to ((\neg\beta \to \alpha) \to \beta)$

Was haben wir in Vorlesung 4 gelernt?

- Wir kennen die drei Axiomenschemata und die Schlussregel modus ponens und wissen, was die Herleitung einer Formel im Frege-Kalkül ist.
- Wir kennen die Relation $\vdash_{\mathsf{Fre}}$ der Frege-Beweisbarkeit.
- Wir kennen das Deduktionstheorem und können es zum Herleiten von Formeln benutzen.
- Wir haben gehört, dass gültige Formeln genau die Frege-beweisbaren Formeln sind, und warten gespannt auf die nächste Vorlesung, in der wir den Beweis dafür sehen werden.

Vorlesung 5:
Die Vollständigkeitssätze für die Kalküle

Einleitung

Ein Kalkül unterteilt die Menge aller Formeln
in beweisbare und nicht-beweisbare Formeln.

Wir vermuten,
dass die beweisbaren Formeln genau die gültigen Formeln sind.
In dieser Vorlesung werden wir beweisen, dass das stimmt.

Unsere Vermutung besteht aus zwei Teilen:

1) Jede beweisbare Formel ist gültig. (Korrektheit des Kalküls)

2) Jede gültige Formel ist beweisbar. (Vollständigkeit des Kalküls)

Die beiden Teile lassen sich am besten getrennt beweisen.
Da wir das für zwei Kalküle machen wollen, sieht es so aus, als ob wir vier (dicke) Beweise führen müssten. Aber: wir können uns einen Beweis sparen!
Wir zeigen die Korrektheit des Frege-Kalküls, die Vollständigkeit des Tableau-Kalküls und wie man Tableau-Beweise in Frege-Beweise umwandeln kann.
Daraus folgt dann die Vollständigkeit des Frege-Kalküls und die Korrektheit des Tableau-Kalküls ...

5.1 Korrektheit des Frege-Kalküls

Wir wollen zeigen, dass jede herleitbare Formel gültig ist,
bzw. allgemeiner,

dass jede Formel, die aus einer Hypothesenmenge Γ herleitbar ist, auch semantische Folgerung der Hypothesenmenge ist – d.h. aus $\Gamma \vdash_{\mathsf{Fre}} \alpha$ folgt $\Gamma \Vdash\!= \alpha$.

Wir werden den Beweis mittels Induktion über die Länge der Herleitung führen.
Jede Herleitung beginnt mit einer Hypothese oder einem Axiom.
Also werden wir benötigen, dass jedes Axiom gültig ist.

Lemma 5.1 (Alle Axiome des Frege-Kalküls sind gültig)

Für alle Formeln α, β und φ gilt

1. $\alpha \to (\beta \to \alpha)$ *ist gültig,*
2. $(\alpha \to (\beta \to \varphi)) \to ((\alpha \to \beta) \to (\alpha \to \varphi))$ *ist gültig und*
3. $\neg\neg\alpha \to \alpha$ *ist gültig.*

Beweis: Sei $\mathcal{A}$ eine Belegung. Wir zeigen $\mathcal{A} \models \gamma$ für jedes Axiom γ.

zu (A1): Es gilt

$$\begin{aligned}
\mathcal{A} \models \alpha \to (\beta \to \alpha) \quad &\Leftrightarrow \quad \mathcal{A} \not\models \alpha \text{ oder } \mathcal{A} \models \beta \to \alpha \\
&\Leftrightarrow \quad \mathcal{A} \not\models \alpha \text{ oder } \mathcal{A} \not\models \beta \text{ oder } \mathcal{A} \models \alpha \\
&\Leftrightarrow \quad \mathcal{A} \not\models \alpha \text{ oder } \mathcal{A} \models \alpha
\end{aligned}$$

Da „$\mathcal{A} \not\models \alpha$ oder $\mathcal{A} \models \alpha$“ eine wahre Aussage ist,
ist „$\mathcal{A} \models \alpha \to (\beta \to \alpha)$“ ebenfalls eine wahre Aussage.

zu (A2): Es gilt

$$\mathcal{A} \models (\alpha \to (\beta \to \varphi)) \to ((\alpha \to \beta) \to (\alpha \to \varphi))$$
$$\Leftrightarrow \quad \mathcal{A} \not\models \alpha \to (\beta \to \varphi) \text{ oder } \mathcal{A} \not\models \alpha \to \beta \text{ oder } \mathcal{A} \not\models \alpha \text{ oder } \mathcal{A} \models \varphi$$
$$\Leftrightarrow \quad \mathcal{A} \not\models \alpha \text{ oder } \mathcal{A} \models \varphi \text{ oder } (\mathcal{A} \models \alpha \text{ und } \mathcal{A} \models \beta \text{ und } \mathcal{A} \not\models \varphi)$$
$$\text{oder } (\mathcal{A} \models \alpha \text{ und } \mathcal{A} \not\models \beta)$$

Da jede Kombination von Erfüllungen von α, β und φ diesen Ausdruck erfüllt, folgt $\mathcal{A} \models (\alpha \to (\beta \to \varphi)) \to ((\alpha \to \beta) \to (\alpha \to \varphi))$.

---

zu (A3): Es gilt

$$\begin{aligned} \mathcal{A} \models \neg\neg\alpha \to \alpha \quad & \Leftrightarrow \quad \mathcal{A} \not\models \neg\neg\alpha \text{ oder } \mathcal{A} \models \alpha \\ & \Leftrightarrow \quad \mathcal{A} \models \neg\alpha \text{ oder } \mathcal{A} \models \alpha \\ & \Leftrightarrow \quad \mathcal{A} \not\models \alpha \text{ oder } \mathcal{A} \models \alpha \qquad (*) \end{aligned}$$

Die Aussage $(*)$ ist wahr.
Folglich ist die äquivalente Aussage „$\mathcal{A} \models \neg\neg\alpha \to \alpha$“ ebenfalls wahr.

Da $\mathcal{A}$ beliebig gewählt wurde,

folgt $\mathcal{A} \models \gamma$ für jede Belegung $\mathcal{A}$ und jedes Axiom γ. ✓

Lemma 5.2 (Korrektheit von $\vdash_{\text{Fre}}$)

Sei α eine Formel und Γ eine Formelmenge.

Aus $\Gamma \vdash_{\text{Fre}} \alpha$ *folgt* $\Gamma \models \alpha$.

(Das verallgemeinert: jedes Theorem des Frege-Kalküls ist gültig.)

Beweis: Induktion über die Länge ℓ der Herleitung von α.

IA $\ell = 1$: Sei α mit einer Herleitung der Länge 1 aus Γ herleitbar.

Dann ist α ein Axiom oder $\alpha \in \Gamma$.

Da jedes Axiom gültig ist (5.1), folgt $\Gamma \models \alpha$.

IV: wenn eine Formel α mit einer Herleitung der Länge $\leqslant k$ aus Γ herleitbar ist, dann folgt $\Gamma \models \alpha$.

IS $\ell = k+1$: α sei mit einer Herleitung der Länge $k+1$ aus Γ herleitbar.

Falls α ein Axiom ist oder $\alpha \in \Gamma$, dann folgt $\Gamma \models \alpha$ (5.1).

Sonst entsteht α mit MP aus α_i und $\alpha_j = \alpha_i \to \alpha$ (mit $i, j \leqslant k$).

Nach IV gilt $\Gamma \models \alpha_i$ und $\Gamma \models \alpha_i \to \alpha$.

Dann gilt auch $\Gamma \models \alpha$ (das hatten wir mal als Übungsaufgabe). ✓

5.2 Vollständigkeit des Tableau-Kalküls

Wir wollen zeigen:
Jede gültige Formel ist Tableau-beweisbar.

D.h.:
Für jede Formel α gilt: wenn α gültig ist, dann ist α Tableau-beweisbar.

Das ist äquivalent zu:
Für jede Formel α gilt:
wenn α nicht Tableau-beweisbar ist, dann ist α nicht gültig.

Anders gesagt:
Für jede Formel α gilt:
wenn jedes geschlossene Tableau für $\neg\alpha$ einen nicht-widersprüchlichen Pfad hat,
dann gibt es eine Belegung $\mathcal{B}$ mit $\mathcal{B} \models \neg\alpha$.

Besser gesagt:
Für jede Formel α gilt:
wenn jedes geschlossene Tableau für α einen nicht-widersprüchlichen Pfad hat,
dann gibt es eine Belegung $\mathcal{B}$ mit $\mathcal{B} \models \alpha$.

Gibt es einen Zusammenhang zwischen nicht-widersprüchlichen Pfaden in geschlossenen Tableaux und erfüllenden Belegungen?

Man betrachtet die Belegung $\mathcal{B}_\pi$ aus
den Atomen auf einem geschlossenen Pfad π
und zeigt, dass diese Belegung alle Formeln auf dem Pfad erfüllt.

Lemma 5.3 (Pfad bestimmt Belegung, die alle seine Formeln erfüllt)

*Sei T ein nicht-widersprüchliches geschlossenes Tableau
und $\pi = \alpha_1, \alpha_2, \ldots, \alpha_k$ ein geschlossener Pfad durch T.
Dann gilt für die Belegung $\mathcal{B}_\pi = \{A_i \mid A_i \in \{\alpha_1, \alpha_2, \ldots, \alpha_k\}\}$
und alle $j = 1, 2, \ldots, k$: $\mathcal{B}_\pi \models \alpha_j$.*

Beweis:

Sei T ein nicht-widersprüchliches geschlossenes Tableau,
und $\pi = \alpha_1, \alpha_2, \ldots, \alpha_k$ sei ein geschlossener Pfad durch T.
(Also ist π nicht widersprüchlich.)

Sei $\mathcal{B}_\pi = \{A_i \mid A_i \in \{\alpha_1, \alpha_2, \ldots, \alpha_k\}\}$ die Belegung,
die genau aus den Atomen auf dem Pfad besteht.

Wir zeigen induktiv für $j = k, k-1, \ldots, 1$, dass $\mathcal{B}_\pi \models \alpha_j$.

IA: z.z.: $\mathcal{B}_\pi \models \alpha_k$.

α_k ist eine Formel, die nicht expandiert werden kann.

Dann ist (1) $\alpha_k = A_i$ mit $A_i \in \mathcal{B}_\pi$,
oder (2) $\alpha_k = \neg A_i$ mit $A_i \notin \mathcal{B}_\pi$ (da π nicht widersprüchlich ist)
oder (3) $\alpha_k = \neg\bot$.
In allen drei Fällen gilt $\mathcal{B}_\pi \models \alpha_k$.
Andere Fälle gibt es nicht.

IV: für bel. festes ℓ und alle $i = k, k-1, \ldots, \ell$ gilt: $\mathcal{B}_\pi \models \alpha_i$.

IS: zu zeigen: $\mathcal{B}_\pi \models \alpha_{\ell-1}$.
Wenn $\alpha_{\ell-1}$ nicht expandiert werden kann, folgt $\mathcal{B}_\pi \models \alpha_{\ell-1}$ wie im IA.
Wenn $\alpha_{\ell-1}$ expandiert werden kann, dann gibt es folgende Möglichkeiten.

Fall 1: $\alpha_{\ell-1} = \neg(\beta \to \gamma)$ mit $\alpha_i = \beta$ und $\alpha_j = \neg\gamma$ für $i, j \geqslant \ell$.

$$\begin{array}{l} \neg(\beta \to \gamma) : \\ \quad\bullet \\ \quad\downarrow \\ \quad\beta \\ \quad\downarrow \\ \quad\neg\gamma \end{array}$$

Da $\mathcal{B}_\pi \models \alpha_i$ und $\mathcal{B}_\pi \models \alpha_j$ (gemäß IV),
folgt $\mathcal{B}_\pi \models \beta$ und $\mathcal{B}_\pi \models \neg\gamma$, also $\mathcal{B}_\pi \models \beta$ und $\mathcal{B}_\pi \not\models \gamma$.
Mit der Semantik von $\to$ folgt daraus $\mathcal{B} \not\models \beta \to \gamma$.
Daraus folgt $\mathcal{B}_\pi \models \neg(\beta \to \gamma)$, d.h. $\mathcal{B} \models \alpha_{\ell-1}$.

Fall 2: $\alpha_{\ell-1} = \beta \to \gamma$ mit $\alpha_i = \neg\beta$ oder $\alpha_i = \gamma$ (und $\gamma \neq \bot$) für ein $i \geqslant \ell$.

$$\begin{array}{ccc} & \beta \to \gamma : & \\ & \bullet & \\ \swarrow & & \searrow \\ \neg\beta & & \gamma \end{array}$$

Da $\mathcal{B}_\pi \models \alpha_i$ (IV), folgt $\mathcal{B}_\pi \models \neg\beta$ oder $\mathcal{B}_\pi \models \gamma$.
Daraus folgt $\mathcal{B}_\pi \not\models \beta$ oder $\mathcal{B}_\pi \models \gamma$.
Mit der Semantik von $\to$ folgt
$\mathcal{B}_\pi \models \beta \to \gamma$, d.h. $\mathcal{B}_\pi \models \alpha_{\ell-1}$.

✓

Lemma 5.4 (Vollständigkeit des Tableau-Kalküls)

Sei α eine Formel und Γ eine endliche Formelmenge.

Aus $\Gamma \mid\!\models \alpha$ folgt $\Gamma \vdash_{\mathsf{Tab}} \alpha$.

(Das verallgemeinert: wenn α gültig ist, dann ist α Tableau-beweisbar.)

Beweis:

Wir zeigen die äquivalente Aussage: wenn $\Gamma \nvdash_{\mathsf{Tab}} \alpha$, dann $\Gamma \mid\!\not\models \alpha$.

Sei $\Gamma \nvdash_{\mathsf{Tab}} \alpha$.

Dann gibt es für $\Gamma \cup \{\neg\alpha\}$ ein geschlossenes Tableau T_0 (3.4),
das nicht widersprüchlich ist.

Also gibt es einen nicht-widersprüchlichen Pfad π durch T_0,
auf dem jeder Knoten expandiert ist.

Nach (5.3) gibt es eine Belegung $\mathcal{B}_\pi$, die jede Formel auf π erfüllt.

Da alle Formeln aus $\Gamma \cup \{\neg\alpha\}$ auf π stehen, folgt $\mathcal{B}_\pi \models \Gamma \cup \{\neg\alpha\}$.

Also gilt $\mathcal{B}_\pi \models \Gamma$ und $\mathcal{B}_\pi \not\models \alpha$.

Das heißt $\Gamma \mid\!\not\models \alpha$. ✓

5.3 Umwandlung von Tableau-Beweisen in Frege-Beweise

Wir wissen, dass jede gültige Formel einen Tableau-Beweis besitzt (5.4).

Wir werden jetzt zeigen,
wie man einen Tableau-Beweis in einen Frege-Beweis umwandelt (5.7).

Damit folgt dann, dass jede gültige Formel einen Frege-Beweis hat (5.9).

Wir wissen auch, dass jeder Frege-Beweis eine gültige Formel liefert (5.2).

Da man jeden Tableau-Beweis in einen Frege-Beweis umwandeln kann (5.7), sind alle Tableau-beweisbaren Formeln ebenfalls gültig (5.11).

Also folgen die Vollständigkeit des Frege-Kalküls und die Korrektheit des Tableau-Kalküls aus unseren bisherigen Ergebnissen und der Möglichkeit, Tableau-Beweise in Frege-Beweise umzuwandeln.
Letzteres werden wir jetzt zeigen.

Frege-Herleitungen aus widersprüchlichen Tableaux

Aus einem widersprüchlichen Tableau für eine Formelmenge Δ
lässt sich eine Frege-Herleitung von $\bot$ aus Δ konstruieren.

Zuerst zeigen wir:
zum Frege-Herleiten von $\bot$ aus einer Hypothesenmenge Γ
sind Formeln, die aus einer Formel in Γ durch Expansion entstehen,
überflüssig.

Lemma 5.5 (Aus $\neg(\beta \to \gamma)$ expandierte Formeln sind überflüssig)

Sei Γ eine Formelmenge, β und γ seien Formeln.

Wenn $\Gamma, \neg(\beta \to \gamma), \beta, \neg\gamma \vdash_{\text{Fre}} \bot$, *dann* $\Gamma, \neg(\beta \to \gamma) \vdash_{\text{Fre}} \bot$.

(Die Beweislänge wächst dabei um eine Konstante unabhängig von Γ, β und γ.)

Beweis:

(1)	$\Gamma, \neg(\beta \to \gamma), \beta, \neg\gamma$	$\vdash_{\text{Fre}} \bot$	Voraussetzung	
(2)	$\Gamma, \neg(\beta \to \gamma), \beta$	$\vdash_{\text{Fre}} \gamma$	(4.9) (1)	
(3)	$\Gamma, \neg(\beta \to \gamma)$	$\vdash_{\text{Fre}} \beta \to \gamma$	DT (2)	
(4)	$\Gamma, \neg(\beta \to \gamma)$	$\vdash_{\text{Fre}} \neg(\beta \to \gamma)$	Hyp	
(5)	$\Gamma, \neg(\beta \to \gamma)$	$\vdash_{\text{Fre}} \bot$	MP (3) (4)	✓

Lemma 5.6 (Aus $\beta \to \gamma$ expandierte Formeln sind überflüssig)

Sei Γ *eine Formelmenge,* β *und* γ *seien Formeln.*

Wenn $\Gamma, \beta \to \gamma, \neg\beta \vdash_{\text{Fre}} \bot$ *und* $\Gamma, \beta \to \gamma, \gamma \vdash_{\text{Fre}} \bot$*, dann* $\Gamma, \beta \to \gamma \vdash_{\text{Fre}} \bot$*.*

(Die Beweislänge wächst dabei um eine Konstante unabhängig von Γ*,* β *und* γ*.)*

Beweis:

(1)	$\Gamma, \beta \to \gamma, \neg\beta$	$\vdash_{\text{Fre}}$	$\bot$	Voraussetzung
(2)	$\Gamma, \beta \to \gamma$	$\vdash_{\text{Fre}}$	β	(4.9) (1)
(3)	$\Gamma, \beta \to \gamma, \gamma$	$\vdash_{\text{Fre}}$	$\bot$	Voraussetzung
(4)	$\Gamma, \beta \to \gamma$	$\vdash_{\text{Fre}}$	$\neg\gamma$	DT (3)
(5)	$\Gamma, \beta \to \gamma$	$\vdash_{\text{Fre}}$	$\beta \to (\neg\gamma \to \neg(\beta \to \gamma))$	(4.13(5))
(6)	$\Gamma, \beta \to \gamma$	$\vdash_{\text{Fre}}$	$\neg\gamma \to \neg(\beta \to \gamma)$	MP (2),(5)
(7)	$\Gamma, \beta \to \gamma$	$\vdash_{\text{Fre}}$	$\neg(\beta \to \gamma)$	MP (4),(6)
(8)	$\Gamma, \beta \to \gamma$	$\vdash_{\text{Fre}}$	$\beta \to \gamma$	Hyp
(9)	$\Gamma, \beta \to \gamma$	$\vdash_{\text{Fre}}$	$\bot$	MP (7),(8) ✓

Lemma 5.7 (Frege-Herleitungen von $\bot$ aus widersprüchlichen Tableaux)

Sei T ein widersprüchliches Tableau für eine endliche Formelmenge Δ. Für jeden Pfad $\pi = \alpha_1, \ldots, \alpha_\ell$ durch T gilt:

für jedes $i = |\Delta|, |\Delta|+1, \ldots, \ell$ ist $\underbrace{\alpha_1, \ldots, \alpha_i}_{\supseteq \Delta} \vdash_{\mathsf{Fre}} \bot$.

Beweis:

Sei m die Länge des längsten Pfades durch T.

Wir führen eine Induktion über $i = m, m-1, \ldots, 1$.

IA: zu zeigen: für alle Pfade $\pi = \alpha_1, \ldots, \alpha_m$ mit maximaler Länge m durch T gilt $\alpha_1, \ldots, \alpha_m \vdash_{\mathsf{Fre}} \bot$.

Da T widersprüchlich ist und π ein längster Pfad durch T ist, ist π widersprüchlich – d.h. es gibt $k, t \leqslant m$ mit $\alpha_k = \neg\alpha_t$.

Mit $\vdash_{\mathsf{Fre}} \underbrace{\neg\alpha_t}_{\alpha_k} \to (\underbrace{\alpha_t \to \bot}_{\neg\alpha_k})$ (4.3) und (DT) folgt $\alpha_k, \alpha_t \vdash_{\mathsf{Fre}} \bot$,

und damit haben wir $\alpha_1, \ldots, \alpha_m \vdash_{\mathsf{Fre}} \bot$.

IV: Für ein $j > |\Delta|$ und
für jeden Pfad $\pi = \alpha_1, \ldots, \alpha_\ell$ durch T gilt $\alpha_1, \ldots, \alpha_{\min\{j,\ell\}} \vdash_{\overline{\text{Fre}}} \perp$.

IS: zu zeigen ist:
Für jeden Pfad $\pi = \alpha_1, \ldots, \alpha_\ell$ durch T gilt $\alpha_1, \ldots, \alpha_{\min\{j-1,\ell\}} \vdash_{\overline{\text{Fre}}} \perp$.

Fall 1: $\ell \leqslant j - 1$,
d.h. π ist ein widersprüchlicher Pfad mit Länge $\leqslant j - 1$.
Dann folgt $\alpha_1, \ldots, \alpha_\ell \vdash_{\overline{\text{Fre}}} \perp$ wie im IA.

Fall 2: $\ell \geqslant j$, d.h. α_j entsteht durch Expansion eines α_e mit $e < j$.

Fall 2a: $\alpha_e = (\beta \to \gamma)$.
Dann gibt es die beiden Pfade $\alpha_1, \ldots, \alpha_e, \ldots, \alpha_{j-1}, \neg\beta, \ldots$
und $\alpha_1, \ldots, \alpha_e, \ldots, \alpha_{j-1}, \gamma, \ldots$ durch T.
Nach IV gilt für deren Anfangsabschnitte der Länge j

$$\alpha_1, \ldots, \alpha_e, \ldots, \alpha_{j-1}, \neg\beta \vdash_{\overline{\text{Fre}}} \perp \text{ und } \alpha_1, \ldots, \alpha_e, \ldots, \alpha_{j-1}, \gamma \vdash_{\overline{\text{Fre}}} \perp.$$

Dann folgt mit Lemma (5.6), dass $\alpha_1, \ldots, \alpha_e, \ldots, \alpha_{j-1} \vdash_{\overline{\text{Fre}}} \perp$.

Fall 2b: $\alpha_e = \neg(\beta \to \gamma)$.

Dann folgt $\alpha_1, \ldots, \alpha_{j-1} \vdash_{\mathsf{Fre}} \bot$ aus der IV und aus Lemma (5.5).

Andere Fälle gibt es nicht. Also gilt stets $\alpha_1, \ldots, \alpha_{j-1} \vdash_{\mathsf{Fre}} \bot$. ✓

Satz 5.8 (aus Tableau-Beweisen können Frege-Beweise gemacht werden)

Sei α eine Formel und Γ eine endliche Formelmenge.

Aus $\Gamma \vdash_{\mathsf{Tab}} \alpha$ folgt $\Gamma \vdash_{\mathsf{Fre}} \alpha$.

Die Länge der Frege-Herleitung von α aus Γ ist dabei asymptotisch höchstens so groß wie ein widersprüchliches Tableau für $\Gamma \cup \{\neg\alpha\}$.

Beweis:

Sei T ein widersprüchliches Tableau für $\Gamma \cup \{\neg\alpha\}$.

Der Fall $i = |\Gamma \cup \{\neg\alpha\}|$ von Lemma (5.7) liefert $\Gamma, \neg\alpha \vdash_{\mathsf{Fre}} \bot$

und daraus folgt $\Gamma \vdash_{\mathsf{Fre}} \alpha$ mit (4.9).

Die Abschätzung der Länge der Herleitung folgt daraus, dass gemäß (5.5)–(5.7) jede Expansion eines Knotens im Tableau durch eine konstante Anzahl an Schritten in der Frege-Herleitung „simuliert“ wird. ✓

Der Vollständigkeitssatz für den Frege-Kalkül

Lemma 5.9 (Vollständigkeitslemma für $\vdash_{\mathsf{Fre}}$)

Sei φ eine Formel und Γ eine endliche Formelmenge.
Wenn $\Gamma \Vdash\!\models \varphi$, dann $\Gamma \vdash_{\mathsf{Fre}} \varphi$.

Beweis:

Gelte $\Gamma \Vdash\!\models \varphi$. Dann folgt $\Gamma \vdash_{\mathsf{Tab}} \varphi$ (5.4). Mit Satz (5.8) folgt $\Gamma \vdash_{\mathsf{Fre}} \varphi$. ✓

Die Korrektheit (5.2) und Vollständigkeit (5.9) liefert den Vollständigkeitssatz für den Frege-Kalkül:

Satz 5.10 (Vollständigkeitssatz für $\vdash_{\mathsf{Fre}}$)

Sei φ eine Formel und Γ eine endliche Formelmenge. Dann gilt:

$$\Gamma \Vdash\!\models \varphi \text{ genau dann, wenn } \Gamma \vdash_{\mathsf{Fre}} \varphi.$$

Der Vollständigkeitssatz für den Tableau-Kalkül

Lemma 5.11 (Korrektheit des Tableau-Kalküls)

Sei φ eine Formel und Γ eine endliche Formelmenge.
Aus $\Gamma \vdash_{\mathsf{Tab}} \varphi$ folgt $\Gamma \Vdash \varphi$.

Beweis:

Gelte $\Gamma \vdash_{\mathsf{Tab}} \varphi$. Mit (5.8) folgt $\Gamma \vdash_{\mathsf{Fre}} \varphi$ und mit (5.2) erhalten wir $\Gamma \Vdash \varphi$. ✓

Die Korrektheit (5.11) und Vollständigkeit (5.4) liefert den Vollständigkeitssatz für den Tableau-Kalkül:

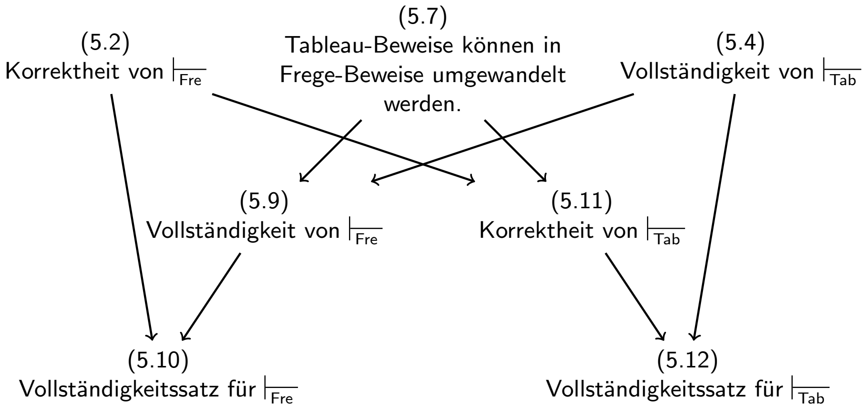
Satz 5.12 (Vollständigkeitssatz für den Tableau-Kalkül)

Sei φ eine Formel und Γ eine endliche Formelmenge. Dann gilt

$$\Gamma \Vdash \varphi \quad \textit{genau dann, wenn} \quad \Gamma \vdash_{\mathsf{Tab}} \alpha.$$

Zusammenfassung

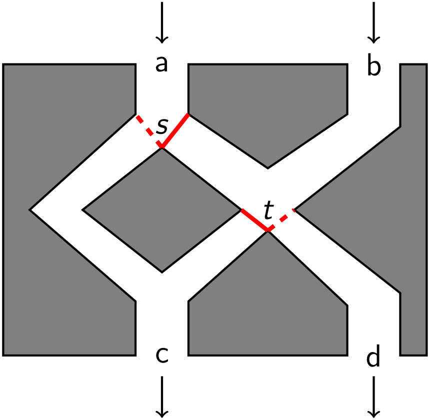
Struktur der wichtigen Ergebnisse von Kapitel 1



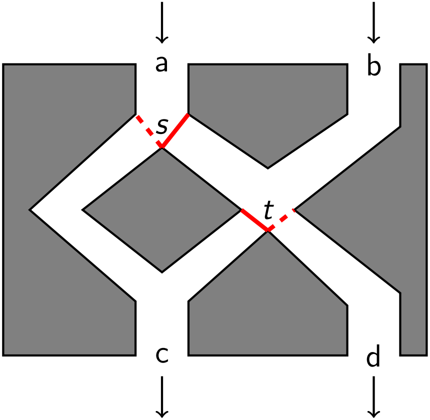
Was haben wir in Vorlesung 5 gelernt?

- Wir kennen die Begriffe Korrektheit, Vollständigkeit und die Vollständigkeitssätze für den Frege-Kalkül und den Tableau-Kalkül.
- Wir wissen, dass die Axiome des Frege-Kalküls gültig sind und können den Beweis für (A1) reproduzieren.
- Wir wissen, dass im Frege-Kalkül nur gültige Formeln herleitbar sind und können den Beweis mittels Induktion über die Länge der Herleitung reproduzieren.
- Wir wissen, dass jede gültige Formel im Tableau-Kalkül bewiesen werden kann und können den Beweis mittels Induktion über die Pfadlänge in geschlossenen Tableaux reproduzieren.
- Wir wissen, wie man Tableau-Beweise in Frege-Beweise umwandeln kann und können den Beweis mittels Induktion über die Pfadlänge in widersprüchlichen Tableaux reproduzieren.
- Wir wissen, wie die Vollständigkeit des Frege-Kalküls aus der Vollständigkeit des Tableau-Kalküls gefolgert werden kann.
- Wir wissen, wie die Korrektheit des Tableau-Kalküls aus der Korrektheit des Frege-Kalküls gefolgert werden kann.

Teil 2: Modale Aussagenlogik

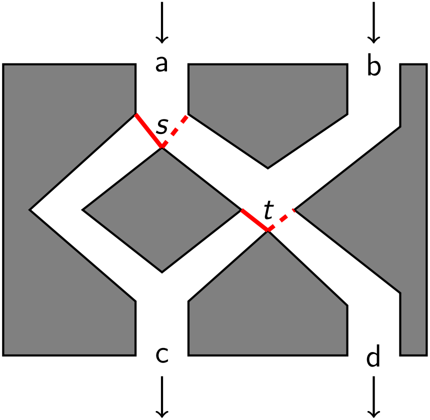


Die Abbildung zeigt eine Murmelbahn mit 2 Eingängen (a und b), 2 Ausgängen (c und d) und 2 Schaltern (s und t).



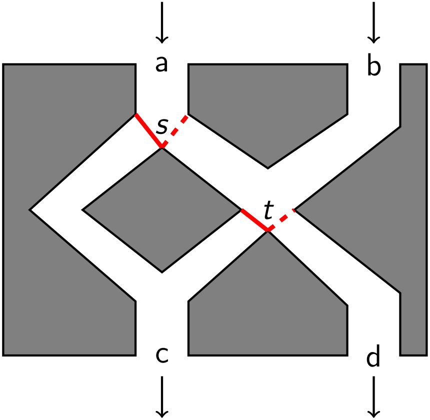
Die Abbildung zeigt eine Murmelbahn mit 2 Eingängen (a und b), 2 Ausgängen (c und d) und 2 Schaltern (s und t).

Startet man eine Murmel in Eingang a, dann läuft sie am Schalter s nach links. Dabei wird der Schalter verändert. Die Murmel fällt bei Ausgang c raus.



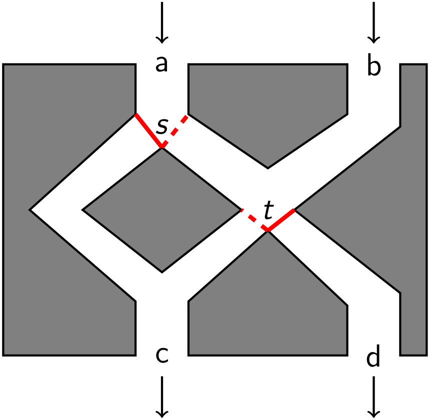
Die Abbildung zeigt eine Murmelbahn mit 2 Eingängen (a und b), 2 Ausgängen (c und d) und 2 Schaltern (s und t).

Startet man eine Murmel in Eingang a, dann läuft sie am Schalter s nach links. Dabei wird der Schalter verändert. Die Murmel fällt bei Ausgang c raus.



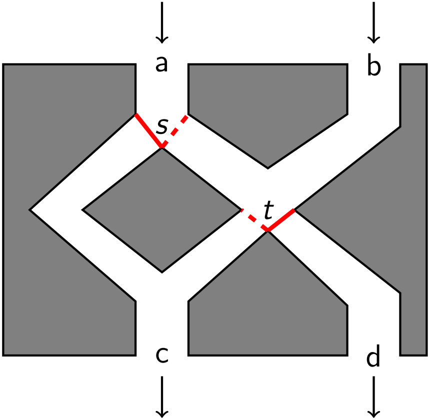
Die Abbildung zeigt eine Murmelbahn mit 2 Eingängen (a und b), 2 Ausgängen (c und d) und 2 Schaltern (s und t).

Startet man nun eine Murmel in Eingang b, dann läuft sie am Schalter t nach rechts. Dabei wird der Schalter verändert. Die Murmel fällt bei Ausgang d raus.



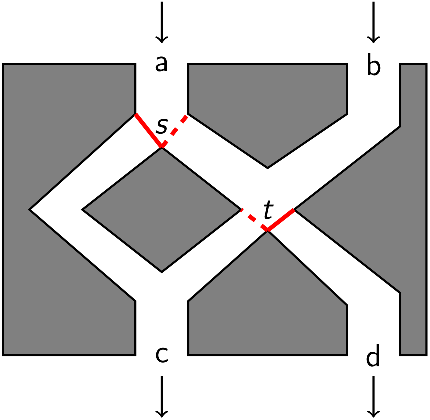
Die Abbildung zeigt eine Murmelbahn mit 2 Eingängen (a und b), 2 Ausgängen (c und d) und 2 Schaltern (s und t).

Startet man nun eine Murmel in Eingang b, dann läuft sie am Schalter t nach rechts. Dabei wird der Schalter verändert. Die Murmel fällt bei Ausgang d raus.

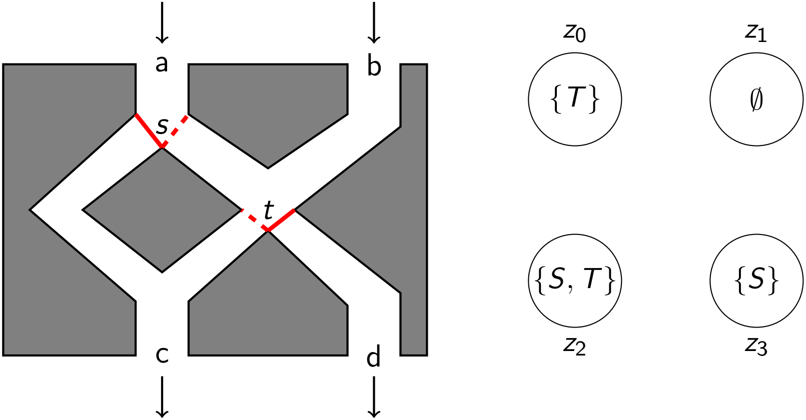


Die Abbildung zeigt eine Murmelbahn mit 2 Eingängen (a und b), 2 Ausgängen (c und d) und 2 Schaltern (s und t).

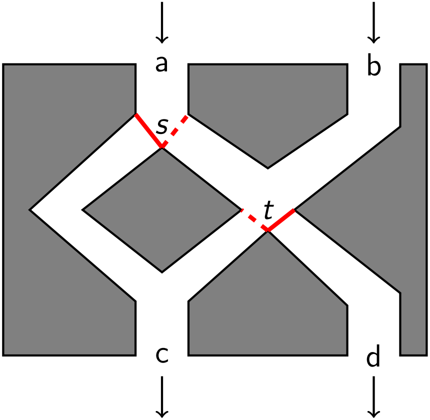
Wir wollen 3 Kugeln in die Kugelbahn werfen.
Kann man – egal wo die erste Kugel reingeworfen wird – die zweite Kugel so werfen, dass die dritte Kugel garantiert bei Ausgang d herauskommt?

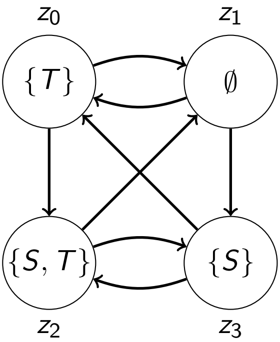


Die Murmelbahn kann 4 Zustände (= Schalterkombinationen) annehmen.
Wir modellieren Schalter s durch Atom S und Schalter t durch Atom T.
Aus Schalterstellungen machen wir Belegungen: eine Belegung enthält ein Atom gdw. der entsprechende Schalter links steht.
Aus möglichen Zustandsübergängen machen wir Kanten.

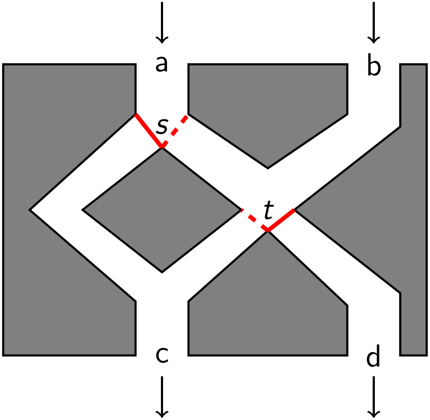


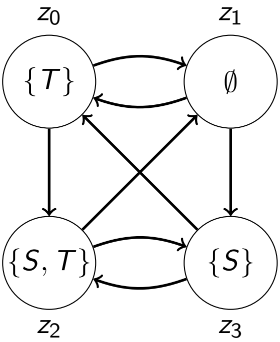
Die Murmelbahn kann 4 Zustände (= Schalterkombinationen) annehmen.
Wir modellieren Schalter s durch Atom S und Schalter t durch Atom T.
Aus Schalterstellungen machen wir Belegungen: eine Belegung enthält ein Atom gdw. der entsprechende Schalter links steht.
Aus möglichen Zustandsübergängen machen wir Kanten.



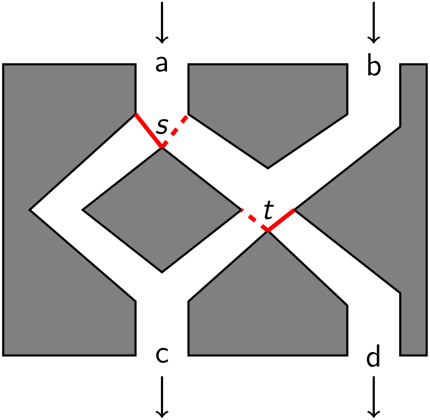


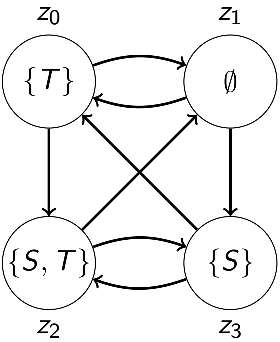
Die Murmelbahn kann 4 Zustände (= Schalterkombinationen) annehmen.
Wir modellieren Schalter s durch Atom S und Schalter t durch Atom T.
Aus Schalterstellungen machen wir Belegungen: eine Belegung enthält ein Atom gdw. der entsprechende Schalter links steht.
Aus möglichen Zustandsübergängen machen wir Kanten.



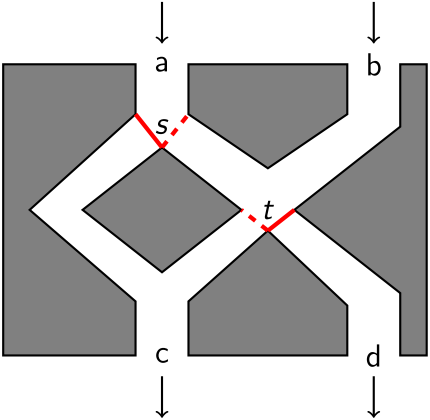


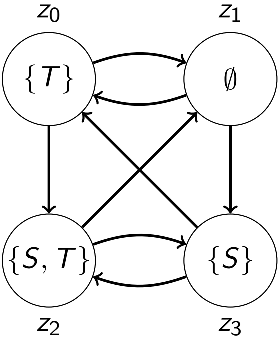
Wir wollen 3 Kugeln in die Kugelbahn werfen.
Kann man – egal wo die erste Kugel reingeworfen wird –
die zweite Kugel so werfen,
dass die dritte Kugel garantiert bei Ausgang d herauskommt?



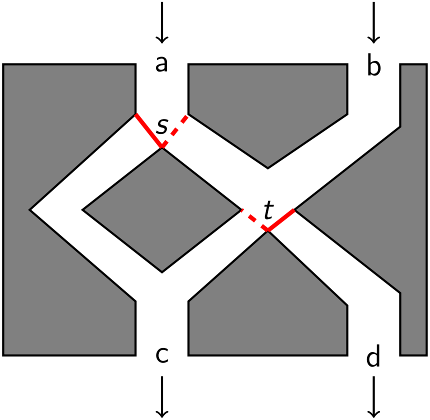


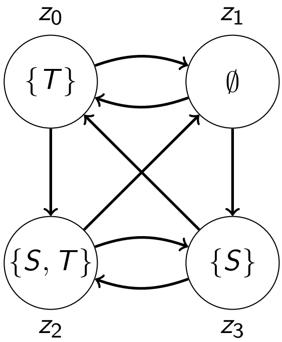
Wir wollen 3 Kugeln in die Kugelbahn werfen.
Kann man – egal wo die erste Kugel reingeworfen wird –
die zweite Kugel so werfen,
dass die dritte Kugel garantiert bei Ausgang d herauskommt? $S \wedge T$





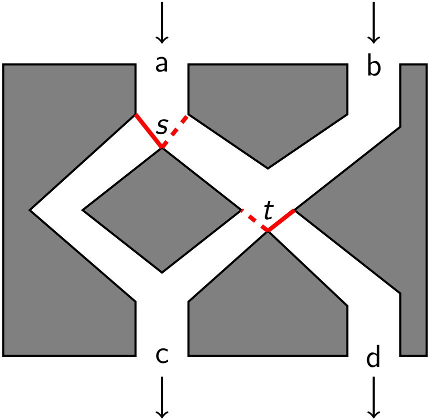
Wir wollen 3 Kugeln in die Kugelbahn werfen.
Kann man – egal wo die erste Kugel reingeworfen wird –
die zweite Kugel so werfen, $\Diamond(S \wedge T)$
dass die dritte Kugel garantiert bei Ausgang d herauskommt? $S \wedge T$

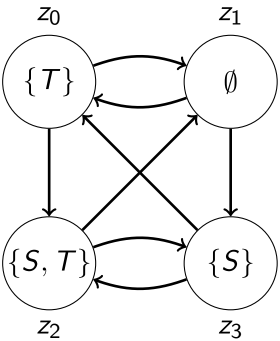




Wir wollen 3 Kugeln in die Kugelbahn werfen.

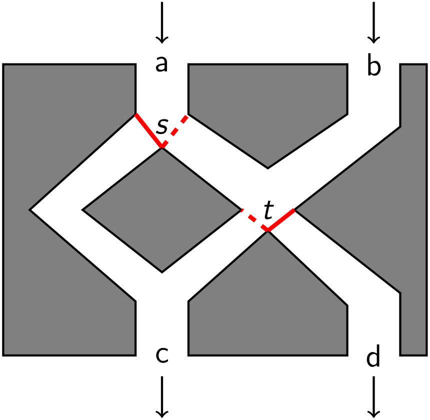
Kann man – egal wo die erste Kugel reingeworfen wird –	$\Box\Diamond(S \wedge T)$
die zweite Kugel so werfen,	$\Diamond(S \wedge T)$
dass die dritte Kugel garantiert bei Ausgang d herauskommt?	$S \wedge T$

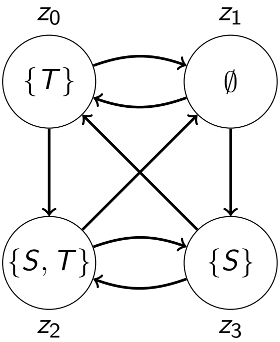




Wir wollen 3 Kugeln in die Kugelbahn werfen.	
Kann man – egal wo die erste Kugel reingeworfen wird –	$\Box\Diamond(S \wedge T)$
die zweite Kugel so werfen,	$\Diamond(S \wedge T)$
dass die dritte Kugel garantiert bei Ausgang d herauskommt?	$S \wedge T$

$z_2 \models S \wedge T$





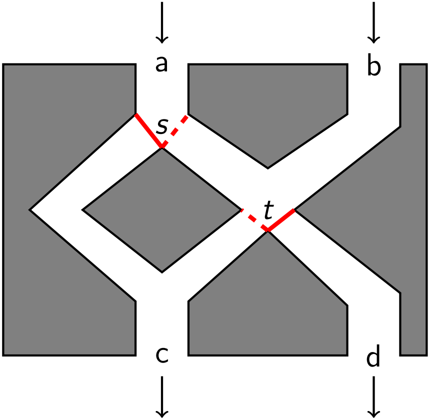
Wir wollen 3 Kugeln in die Kugelbahn werfen.

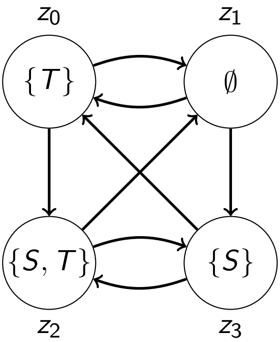
Kann man – egal wo die erste Kugel reingeworfen wird – $\quad \Box\Diamond(S \wedge T)$

die zweite Kugel so werfen, $\quad \Diamond(S \wedge T)$

dass die dritte Kugel garantiert bei Ausgang d herauskommt? $\quad S \wedge T$

$z_2 \models S \wedge T$, $z_0 \models \Diamond(S \wedge T)$, $z_3 \models \Diamond(S \wedge T)$





Wir wollen 3 Kugeln in die Kugelbahn werfen.	
Kann man – egal wo die erste Kugel reingeworfen wird –	$\Box\Diamond(S \wedge T)$
die zweite Kugel so werfen,	$\Diamond(S \wedge T)$
dass die dritte Kugel garantiert bei Ausgang d herauskommt?	$S \wedge T$

$z_2 \models S \wedge T$, $z_0 \models \Diamond(S \wedge T)$, $z_3 \models \Diamond(S \wedge T)$, $z_1 \models \Box\Diamond(S \wedge T)$

Teil 2: Modale Aussagenlogik

Während in der bisher betrachteten *klassischen* Aussagenlogik nur die Wahrheit von Aussagen modelliert wird, spielt in der modalen Aussagenlogik die Art und Weise (*modus*), in der ein Aussage wahr ist, eine Rolle.

Eine Aussage kann

- notwendigerweise oder möglicherweise wahr sein,
- heute, gestern oder morgen wahr sein,
- geglaubt oder gewusst werden,
- vor/nach der Ausführung eines Programmschrittes wahr sein.

Wir werden die Grundlagen der Modallogik mit zwei Beweiskalkülen sowie Algorithmen betrachten.

Teil 2: Modale Aussagenlogik

2. Modale Aussagenlogik

[Literatur (siehe Semesterapparat):

Kreuzer, Kühling: Logik für Informatiker	VL06/07
Priest: An Introduction to Non-classical Logic	VL06/07
Nerode, Shore: Logic for Applications	VL06/07
Halpern, Moses: A guide to completeness and complexity of modal logic	VL08

]

Vorlesung 6:
Grundbegriffe der modalen Aussagenlogik

Wir werden die Grundbegriffe der Modallogik (bzw. modalen Aussagenlogik) definieren.
Dazu erweitern wir die Begriffe, die wir bereits für die Aussagenlogik kennen.

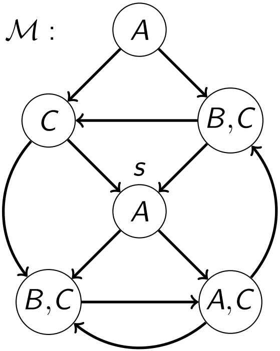
- Wie sehen modallogische Formeln aus?
- Was ist ein Kripke-Modell und wann erfüllt es eine Formel?
- Was sind gültige, erfüllbare und unerfüllbare Formeln?
- Wann sind Formeln äquivalent?
- Welche Verknüpfungszeichen braucht man überhaupt in Formeln? (adäquate Verknüpfungszeichen)

Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\quad \Diamond(A \vee \neg B)$, $\quad \Box(\Diamond(\Box A \wedge \Diamond\neg\Box(B \vee C)) \wedge A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)
(die Darstellung ist ohne Mengenklammern)



$\mathcal{M}$ erfüllt die Formel $\Box\varphi$ in Welt s
genau dann, wenn
alle Nachfolger von s Formel φ erfüllen.

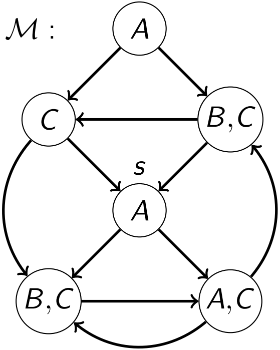
Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\quad \Diamond(A \vee \neg B)$, $\quad \Box(\Diamond(\Box A \wedge \Diamond\neg\Box(B \vee C)) \wedge A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)
(die Darstellung ist ohne Mengenklammern)

$\mathcal{M}$:



$\mathcal{M}$ erfüllt die Formel $\Box\varphi$ in Welt s
genau dann, wenn
alle Nachfolger von s Formel φ erfüllen.

$$\mathcal{M}, s \models_K \quad A$$
$$\mathcal{M}, s \not\models_K \quad A \wedge C$$

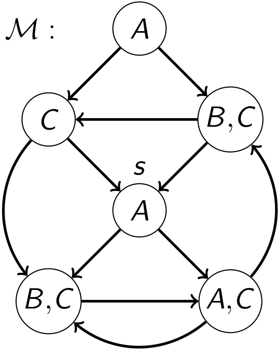
Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\quad \Diamond(A \lor \neg B)$, $\quad \Box(\Diamond(\Box A \land \Diamond\neg\Box(B \lor C)) \land A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)
(die Darstellung ist ohne Mengenklammern)

$\mathcal{M}$:



$\mathcal{M}$ erfüllt die Formel $\Box\varphi$ in Welt s
genau dann, wenn
alle Nachfolger von s Formel φ erfüllen.

$$\mathcal{M}, s \models_K A$$
$$\mathcal{M}, s \not\models_K A \land C$$
$$\mathcal{M}, s \models_K \Box C$$

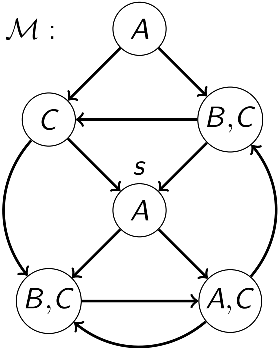
Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\quad \Diamond(A \vee \neg B)$, $\quad \Box(\Diamond(\Box A \wedge \Diamond\neg\Box(B \vee C)) \wedge A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)
(die Darstellung ist ohne Mengenklammern)

$\mathcal{M}$:



$\mathcal{M}$ erfüllt die Formel $\Box\varphi$ in Welt s
genau dann, wenn
alle Nachfolger von s Formel φ erfüllen.

$$\mathcal{M}, s \models_K A$$
$$\mathcal{M}, s \not\models_K A \wedge C$$
$$\mathcal{M}, s \models_K \Box C$$
$$\mathcal{M}, s \models_K \Box(B \vee \Box C)$$

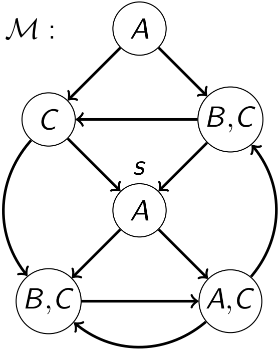
Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\quad \Diamond(A \vee \neg B)$, $\quad \Box(\Diamond(\Box A \wedge \Diamond\neg\Box(B \vee C)) \wedge A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)
(die Darstellung ist ohne Mengenklammern)

$\mathcal{M}$:



$\mathcal{M}$ erfüllt die Formel $\Box\varphi$ in Welt s
genau dann, wenn
alle Nachfolger von s Formel φ erfüllen.

$$\mathcal{M}, s \models_K \quad A$$
$$\mathcal{M}, s \not\models_K \quad A \wedge C$$
$$\mathcal{M}, s \models_K \quad \Box C$$
$$\mathcal{M}, s \models_K \quad \Box(B \vee \Box C)$$
$$\mathcal{M}, s \not\models_K \quad \Box B$$

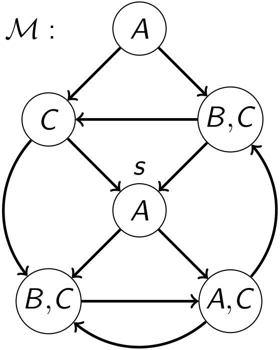
Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\quad \Diamond(A \vee \neg B)$, $\quad \Box(\Diamond(\Box A \wedge \Diamond\neg\Box(B \vee C)) \wedge A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)
(die Darstellung ist ohne Mengenklammern)

$\mathcal{M}$:



$\mathcal{M}$ erfüllt die Formel $\Box\varphi$ in Welt s
genau dann, wenn
alle Nachfolger von s Formel φ erfüllen.

$$\mathcal{M}, s \models_K A$$
$$\mathcal{M}, s \not\models_K A \wedge C$$
$$\mathcal{M}, s \models_K \Box C$$
$$\mathcal{M}, s \models_K \Box(B \vee \Box C)$$
$$\mathcal{M}, s \not\models_K \Box B$$
$$\mathcal{M}, s \models_K \neg\Box B$$

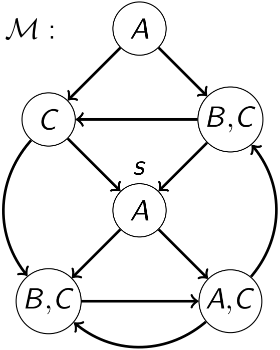
Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\quad \Diamond(A \vee \neg B)$, $\quad \Box(\Diamond(\Box A \wedge \Diamond\neg\Box(B \vee C)) \wedge A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)
(die Darstellung ist ohne Mengenklammern)

$\mathcal{M}$:



$\mathcal{M}$ erfüllt die Formel $\Box\varphi$ in Welt s
genau dann, wenn
alle Nachfolger von s Formel φ erfüllen.

$$\mathcal{M}, s \models_K A$$
$$\mathcal{M}, s \not\models_K A \wedge C$$
$$\mathcal{M}, s \models_K \Box C$$
$$\mathcal{M}, s \models_K \Box(B \vee \Box C)$$
$$\mathcal{M}, s \not\models_K \Box B$$
$$\mathcal{M}, s \models_K \neg\Box B$$
$$\mathcal{M}, s \models_K \Box C \wedge \neg\Box B$$

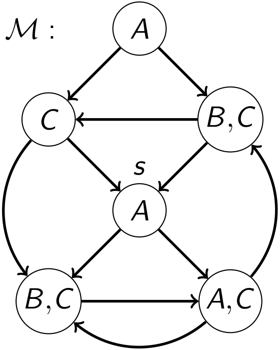
Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\quad \Diamond(A \vee \neg B)$, $\quad \Box(\Diamond(\Box A \wedge \Diamond\neg\Box(B \vee C)) \wedge A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)
(die Darstellung ist ohne Mengenklammern)

$\mathcal{M}$:



$\mathcal{M}$ erfüllt die Formel $\Box\varphi$ in Welt s
genau dann, wenn
alle Nachfolger von s Formel φ erfüllen.

$$\begin{array}{ll}
\mathcal{M}, s \models_K & A \\
\mathcal{M}, s \not\models_K & A \wedge C \\
\mathcal{M}, s \models_K & \Box C \\
\mathcal{M}, s \models_K & \Box(B \vee \Box C) \\
\mathcal{M}, s \not\models_K & \Box B \\
\mathcal{M}, s \models_K & \neg\Box B \\
\mathcal{M}, s \models_K & \Box C \wedge \neg\Box B \\
\mathcal{M}, s \not\models_K & \Box\Box\Box A
\end{array}$$

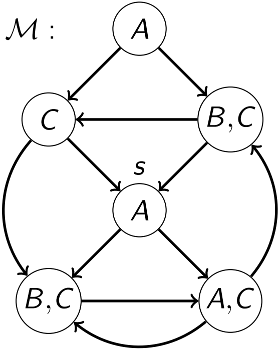
Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\Diamond(A \vee \neg B)$, $\Box(\Diamond(\Box A \wedge \Diamond\neg\Box(B \vee C)) \wedge A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)

$\mathcal{M}$:



$\mathcal{M}$ erfüllt $\Diamond\varphi$ in Welt s
genau dann, wenn
mind. ein Nachfolger von s erfüllt φ

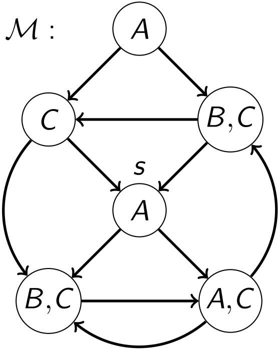
Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\quad \Diamond(A \vee \neg B)$, $\quad \Box(\Diamond(\Box A \wedge \Diamond\neg\Box(B \vee C)) \wedge A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)

$\mathcal{M}$:



$\mathcal{M}$ erfüllt $\Diamond\varphi$ in Welt s
genau dann, wenn
mind. ein Nachfolger von s erfüllt φ

$\mathcal{M}, s \models_K \quad \Diamond B$

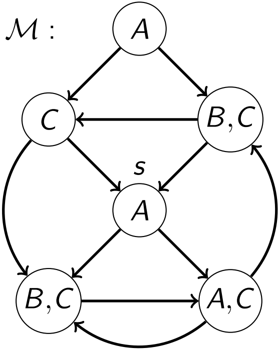
Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\quad \Diamond(A \vee \neg B)$, $\quad \Box(\Diamond(\Box A \wedge \Diamond\neg\Box(B \vee C)) \wedge A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)

$\mathcal{M}$:



$\mathcal{M}$ erfüllt $\Diamond\varphi$ in Welt s
genau dann, wenn
mind. ein Nachfolger von s erfüllt φ

$\mathcal{M}, s \models_K \quad \Diamond B$

$\mathcal{M}, s \models_K \quad \Diamond A$

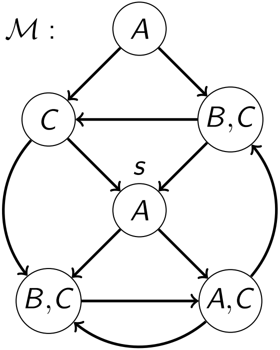
Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\quad \Diamond(A \vee \neg B)$, $\quad \Box(\Diamond(\Box A \wedge \Diamond\neg\Box(B \vee C)) \wedge A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)

$\mathcal{M}$:



$\mathcal{M}$ erfüllt $\Diamond\varphi$ in Welt s
genau dann, wenn
mind. ein Nachfolger von s erfüllt φ

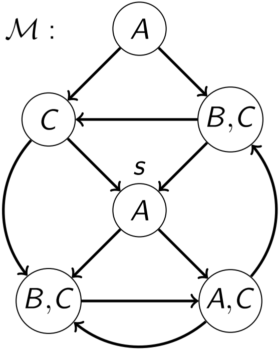
$$\mathcal{M}, s \models_K \Diamond B$$
$$\mathcal{M}, s \models_K \Diamond A$$
$$\mathcal{M}, s \not\models_K \Diamond(A \wedge B)$$

Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\quad \Diamond(A \vee \neg B)$, $\quad \Box(\Diamond(\Box A \wedge \Diamond\neg\Box(B \vee C)) \wedge A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)



$\mathcal{M}$ erfüllt $\Diamond\varphi$ in Welt s
genau dann, wenn
mind. ein Nachfolger von s erfüllt φ

$$\mathcal{M}, s \models_K \Diamond B$$
$$\mathcal{M}, s \models_K \Diamond A$$
$$\mathcal{M}, s \not\models_K \Diamond(A \wedge B)$$
$$\mathcal{M}, s \models_K \Diamond\Diamond(A \wedge C)$$

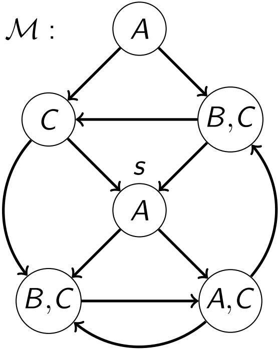
Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\quad \Diamond(A \vee \neg B)$, $\quad \Box(\Diamond(\Box A \wedge \Diamond\neg\Box(B \vee C)) \wedge A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)

$\mathcal{M}$:



$\mathcal{M}$ erfüllt $\Diamond\varphi$ in Welt s
genau dann, wenn
mind. ein Nachfolger von s erfüllt φ

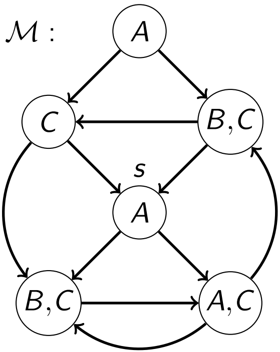
$$\mathcal{M}, s \models_K \Diamond B$$
$$\mathcal{M}, s \models_K \Diamond A$$
$$\mathcal{M}, s \not\models_K \Diamond(A \wedge B)$$
$$\mathcal{M}, s \models_K \Diamond\Diamond(A \wedge C)$$
$$\mathcal{M}, s \models_K \Diamond\Diamond\Box(\neg A \wedge B)$$

Einführendes Beispiel zur Modallogik

Modallogische Formeln haben zusätzliche einstellige Operatoren: $\Box$ und $\Diamond$.

Formelbeispiele: $\Box A$, $\quad \Diamond(A \vee \neg B)$, $\quad \Box(\Diamond(\Box A \wedge \Diamond\neg\Box(B \vee C)) \wedge A)$

Semantik: Graph $\mathcal{M}$ aus Belegungen (=Kripke-Modell)



$\mathcal{M}$ erfüllt $\Diamond\varphi$ in Welt s
genau dann, wenn
mind. ein Nachfolger von s erfüllt φ

$$
\begin{aligned}
\mathcal{M}, s &\models_K \quad \Diamond B \\
\mathcal{M}, s &\models_K \quad \Diamond A \\
\mathcal{M}, s &\not\models_K \quad \Diamond(A \wedge B) \\
\mathcal{M}, s &\models_K \quad \Diamond\Diamond(A \wedge C) \\
\mathcal{M}, s &\models_K \quad \Diamond\Diamond\Box(\neg A \wedge B) \\
\mathcal{M}, s &\models_K \quad \neg\Box\neg B
\end{aligned}
$$

Modallogik formal: Formeln

Definition 6.1 (modallogische Formel)

Eine atomare Formel hat die Form A_i für $i = 0, 1, 2, \ldots$

Modallogische Formeln sind induktiv definiert wie folgt.

1. Die Konstanten $\top$ und $\bot$ und alle atomaren Formeln sind Formeln.
2. - Für jede Formel α ist $\neg\alpha$ eine Formel.
   - Für jede Formel α sind $\Box\alpha$ sowie $\Diamond\alpha$ Formeln.
   - Für alle Formeln α und β sind $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$ sowie $(\alpha \to \beta)$ ebenfalls Formeln.

„$\Box\alpha$“ spricht man „Box α“ und „$\Diamond\alpha$“ spricht man „Diamant α“ aus.

Beispiel:

$$\Diamond\neg\Diamond(A_1 \vee \Box\Box(A_3 \wedge (\Box A_3 \wedge A_5)))$$

Modallogik formal: (Kripke-)Semantik

Definition 6.2 (Kripke-Modell)

Ein Kripke-Modell $\mathcal{M} = (W, R, \xi)$ besteht aus

- einem gerichteten Graph (W, R)
  mit Knotenmenge W (*Welten*), $W \neq \emptyset$,
  Kantenmenge $R \subseteq W \times W$, und
- einer Belegung $\xi(w) \subseteq \{A_0, A_1, A_2, \ldots\}$ für jede Welt $w \in W$
  (d.h. $\xi : W \to \mathcal{P}(\{A_0, A_1, A_2, \ldots\})$).

Konkrete Kripke-Modelle werden gerne graphisch angegeben.

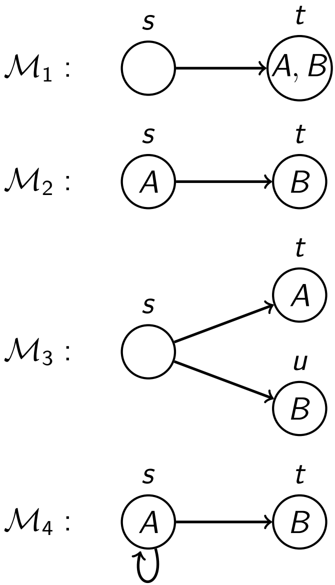
Definition 6.3 (modallogische Erfüllungsrelation $\models_K$)

Sei $\mathcal{M} = (W, R, \xi)$ ein Kripke-Modell,
$w \in W$ sei eine Welt von $\mathcal{M}$, α und β seien modallogische Formeln.
Die Erfüllungsrelation $\models_K$ zwischen Modellen, Welten und Formeln ist wie folgt definiert.

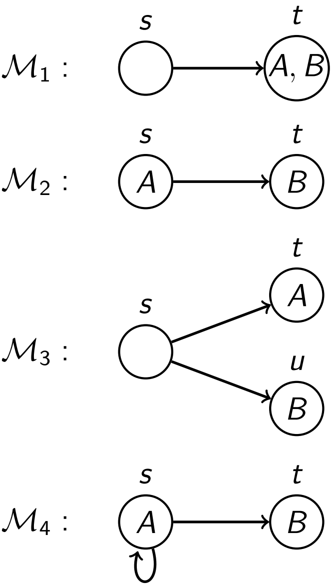
$\mathcal{M}, w \models_K \top$		
$\mathcal{M}, w \not\models_K \bot$		
$\mathcal{M}, w \models_K A_i$	gdw.	$A_i \in \xi(w)$
$\mathcal{M}, w \models_K \neg\alpha$	gdw.	$\mathcal{M}, w \not\models_K \alpha$
$\mathcal{M}, w \models_K \alpha \wedge \beta$	gdw.	$\mathcal{M}, w \models_K \alpha$ und $\mathcal{M}, w \models_K \beta$
$\mathcal{M}, w \models_K \alpha \vee \beta$	gdw.	$\mathcal{M}, w \models_K \alpha$ oder $\mathcal{M}, w \models_K \beta$
$\mathcal{M}, w \models_K \alpha \to \beta$	gdw.	$\mathcal{M}, w \not\models_K \alpha$ oder $\mathcal{M}, w \models_K \beta$
$\mathcal{M}, w \models_K \Box\,\alpha$	gdw.	für alle $t \in W$: wenn $(w, t) \in R$, dann $\mathcal{M}, t \models_K \alpha$
$\mathcal{M}, w \models_K \Diamond\,\alpha$	gdw.	es gibt ein $t \in W$: $(w, t) \in R$ und $\mathcal{M}, t \models_K \alpha$

„$\mathcal{M}, w \models_K \varphi$“ spricht man „$\varphi$ ist in Welt w von Modell $\mathcal{M}$ erfüllt“ aus.

Beispiele

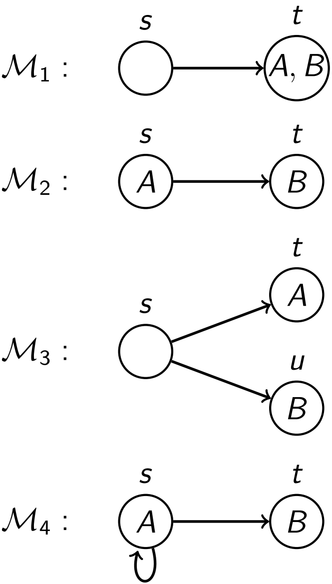

$\mathcal{M}_1$:
s
t
A, B
$\mathcal{M}_2$:
s
t
A
B
$\mathcal{M}_3$:
s
t
u
A
B
$\mathcal{M}_4$:
s
t
A
B

Beispiele



$$\mathcal{M}_1, s \models_K \Diamond A \land \Diamond B$$

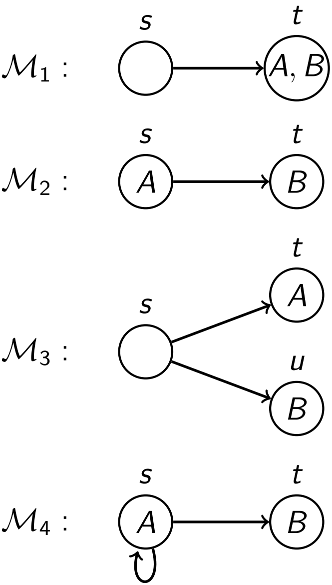
Beispiele



$$\mathcal{M}_1, s \models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_2, s \not\models_K \Diamond A \wedge \Diamond B$$

Beispiele

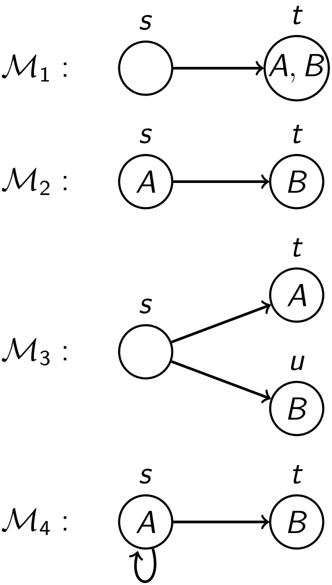


$$\mathcal{M}_1, s \underset{K}{\models} \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_2, s \underset{K}{\not\models} \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_3, s \underset{K}{\models} \Diamond A \wedge \Diamond B$$

Beispiele



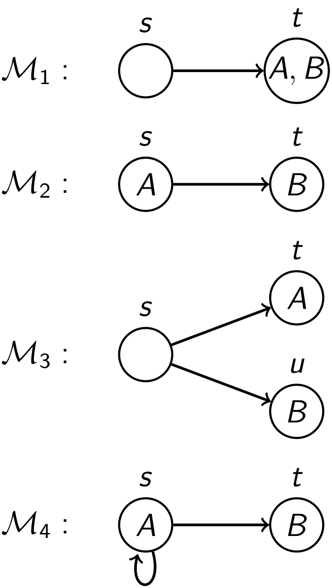
$$\mathcal{M}_1, s \models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_2, s \not\models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_3, s \models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_4, s \models_K \Diamond A \wedge \Diamond B$$

Beispiele



$$\mathcal{M}_1, s \models_K \Diamond A \wedge \Diamond B$$

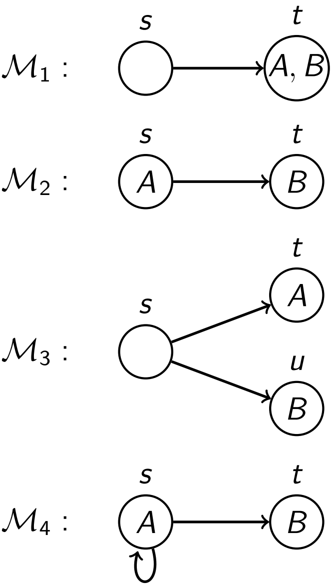
$$\mathcal{M}_2, s \not\models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_3, s \models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_4, s \models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_1, s \models_K \Box A \wedge \Box B$$

Beispiele



$$\mathcal{M}_1, s \models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_2, s \not\models_K \Diamond A \wedge \Diamond B$$

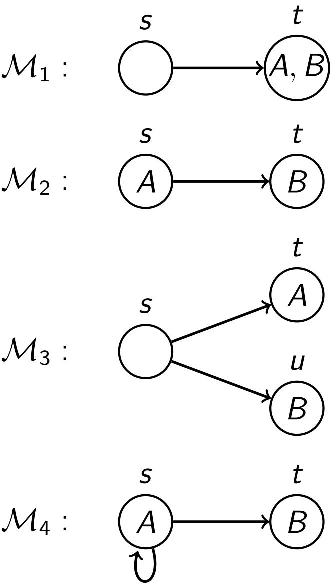
$$\mathcal{M}_3, s \models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_4, s \models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_1, s \models_K \Box A \wedge \Box B$$

$$\mathcal{M}_2, s \models_K \Box A \vee \Box B$$

Beispiele



$$\mathcal{M}_1, s \models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_2, s \not\models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_3, s \models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_4, s \models_K \Diamond A \wedge \Diamond B$$

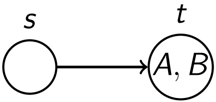
$$\mathcal{M}_1, s \models_K \Box A \wedge \Box B$$

$$\mathcal{M}_2, s \models_K \Box A \vee \Box B$$

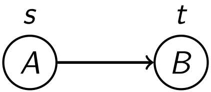
$$\mathcal{M}_3, s \not\models_K \Box A \vee \Box B$$

Beispiele

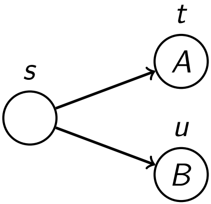
$\mathcal{M}_1$:



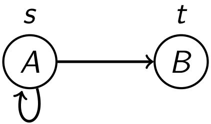
$\mathcal{M}_2$:



$\mathcal{M}_3$:



$\mathcal{M}_4$:



$$\mathcal{M}_1, s \models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_2, s \not\models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_3, s \models_K \Diamond A \wedge \Diamond B$$

$$\mathcal{M}_4, s \models_K \Diamond A \wedge \Diamond B$$

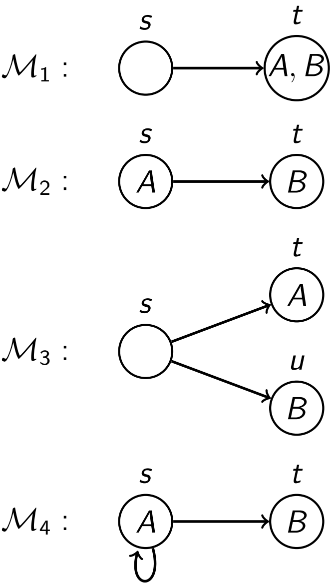
$$\mathcal{M}_1, s \models_K \Box A \wedge \Box B$$

$$\mathcal{M}_2, s \models_K \Box A \vee \Box B$$

$$\mathcal{M}_3, s \not\models_K \Box A \vee \Box B$$

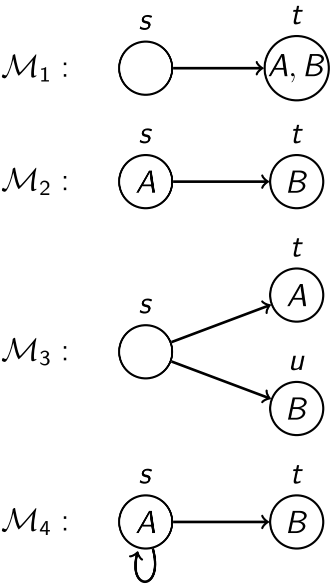
$$\mathcal{M}_4, s \not\models_K \Box A \vee \Box B$$

Beispiele



$$\mathcal{M}_3, s \models_K \Diamond A \wedge \Diamond \neg A$$

Beispiele

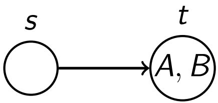


$$\mathcal{M}_3, s \models_K \Diamond A \land \Diamond \neg A$$

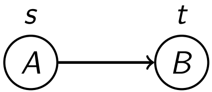
$$\mathcal{M}_3, s \not\models_K \Diamond (A \land \neg A)$$

Beispiele

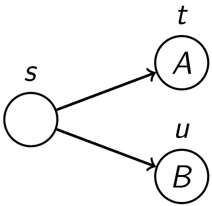
$\mathcal{M}_1$:



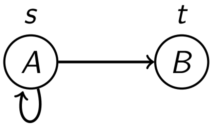
$\mathcal{M}_2$:



$\mathcal{M}_3$:



$\mathcal{M}_4$:

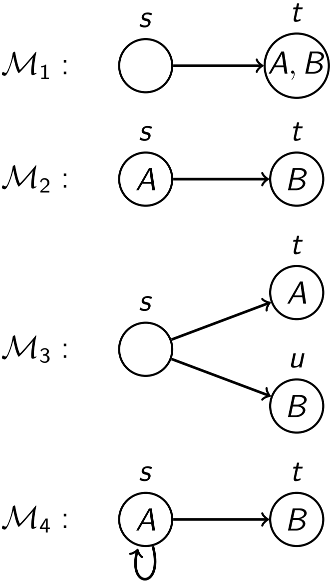


$$\mathcal{M}_3, s \models_K \Diamond A \land \Diamond \neg A$$

$$\mathcal{M}_3, s \not\models_K \Diamond (A \land \neg A)$$

$$\mathcal{M}_3, s \not\models_K \Box A \land \Box \neg A$$

Beispiele



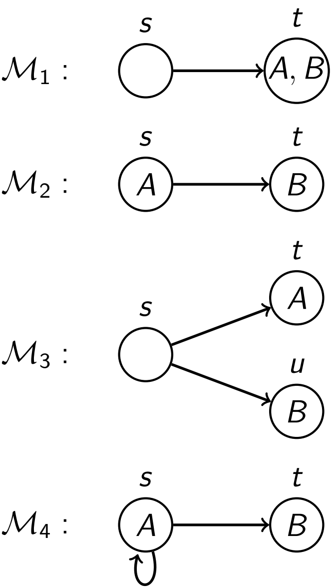
$$\mathcal{M}_3, s \models_K \Diamond A \wedge \Diamond \neg A$$

$$\mathcal{M}_3, s \not\models_K \Diamond (A \wedge \neg A)$$

$$\mathcal{M}_3, s \not\models_K \Box A \wedge \Box \neg A$$

$$\mathcal{M}_3, t \models_K \Box A \wedge \Box \neg A$$

Beispiele



$$\mathcal{M}_3, s \models_K \Diamond A \land \Diamond \neg A$$

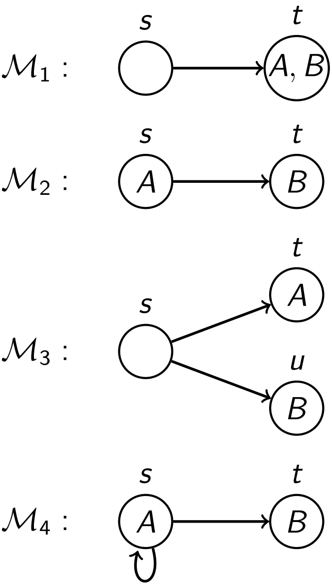
$$\mathcal{M}_3, s \not\models_K \Diamond (A \land \neg A)$$

$$\mathcal{M}_3, s \not\models_K \Box A \land \Box \neg A$$

$$\mathcal{M}_3, t \models_K \Box A \land \Box \neg A$$

$$\mathcal{M}_3, t \models_K \Box \bot$$

Beispiele



$$\mathcal{M}_3, s \models_K \Diamond A \wedge \Diamond\neg A$$

$$\mathcal{M}_3, s \not\models_K \Diamond(A \wedge \neg A)$$

$$\mathcal{M}_3, s \not\models_K \Box A \wedge \Box\neg A$$

$$\mathcal{M}_3, t \models_K \Box A \wedge \Box\neg A$$

$$\mathcal{M}_3, t \models_K \Box\bot$$

$$\mathcal{M}_3, t \not\models_K \Diamond\top$$

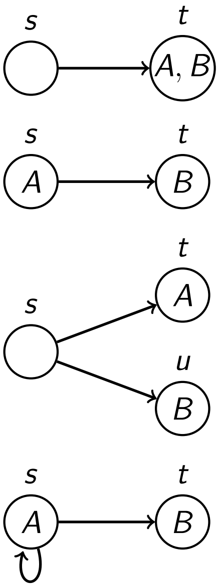
Beispiele

$\mathcal{M}_1$:

$\mathcal{M}_2$:

$\mathcal{M}_3$:

$\mathcal{M}_4$:



$$\mathcal{M}_3, s \models_K \Diamond A \wedge \Diamond \neg A$$

$$\mathcal{M}_3, s \not\models_K \Diamond (A \wedge \neg A)$$

$$\mathcal{M}_3, s \not\models_K \Box A \wedge \Box \neg A$$

$$\mathcal{M}_3, t \models_K \Box A \wedge \Box \neg A$$

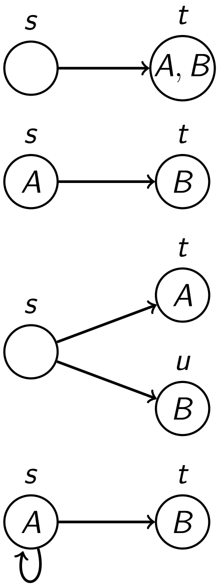
$$\mathcal{M}_3, t \models_K \Box \bot$$

$$\mathcal{M}_3, t \not\models_K \Diamond \top$$

$$\mathcal{M}_3, s \models_K \Diamond \top$$

Beispiele

$\mathcal{M}_1$: $\mathcal{M}_2$: $\mathcal{M}_3$: $\mathcal{M}_4$:



$$\mathcal{M}_3, s \models_K \Diamond A \wedge \Diamond \neg A$$

$$\mathcal{M}_3, s \not\models_K \Diamond (A \wedge \neg A)$$

$$\mathcal{M}_3, s \not\models_K \Box A \wedge \Box \neg A$$

$$\mathcal{M}_3, t \models_K \Box A \wedge \Box \neg A$$

$$\mathcal{M}_3, t \models_K \Box \bot$$

$$\mathcal{M}_3, t \not\models_K \Diamond \top$$

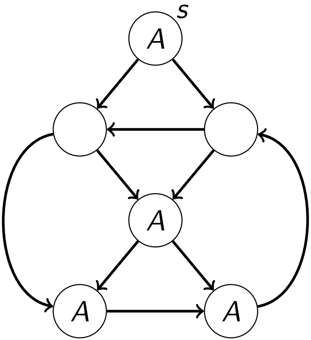
$$\mathcal{M}_3, s \models_K \Diamond \top$$

$$\mathcal{M}_3, s \not\models_K \Box \bot$$

Formelauswertung

mit Dynamischem Programmieren

für alle Teilformeln α in aufsteigender Reihenfolge wiederhole:
markiere jede Welt, die α erfüllt ...



entscheide $\mathcal{M}, s \models_K \Box\Diamond\Box A$:

1.) markiere mit $\Box A$

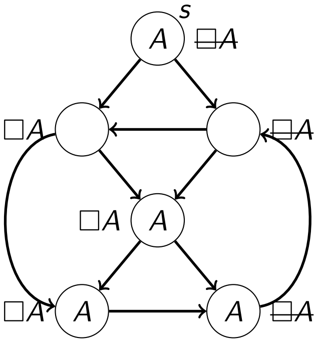
2.) markiere mit $\Diamond\Box A$

3.) entscheide $\mathcal{M}, s \models_K \Box\Diamond\Box A$

Formelauswertung

mit Dynamischem Programmieren

für alle Teilformeln α in aufsteigender Reihenfolge wiederhole:
  markiere jede Welt, die α erfüllt ...



entscheide $\mathcal{M}, s \models_K \Box\Diamond\Box A$:

1.) markiere mit $\Box A$

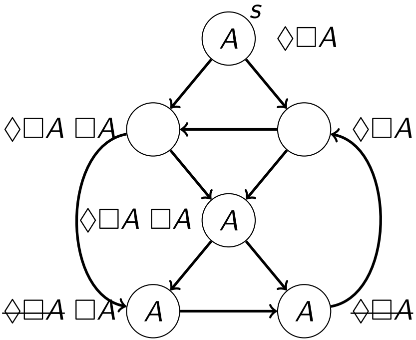
2.) markiere mit $\Diamond\Box A$

3.) entscheide $\mathcal{M}, s \models_K \Box\Diamond\Box A$

Formelauswertung

mit Dynamischem Programmieren

für alle Teilformeln α in aufsteigender Reihenfolge wiederhole:
markiere jede Welt, die α erfüllt ...



entscheide $\mathcal{M}, s \models_K \Box\Diamond\Box A$:

1.) markiere mit $\Box A$

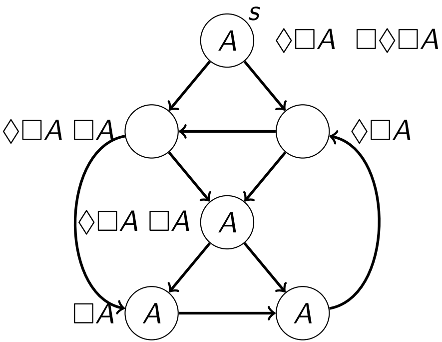
2.) markiere mit $\Diamond\Box A$

3.) entscheide $\mathcal{M}, s \models_K \Box\Diamond\Box A$

Formelauswertung

mit Dynamischem Programmieren

für alle Teilformeln α in aufsteigender Reihenfolge wiederhole:
markiere jede Welt, die α erfüllt ...



entscheide $\mathcal{M}, s \models_K \Box\Diamond\Box A$:

1.) markiere mit $\Box A$

2.) markiere mit $\Diamond\Box A$

3.) entscheide $\mathcal{M}, s \models_K \Box\Diamond\Box A$

Rechenzeit: Anzahl Teilformeln $\cdot$ Anz.Welten $\cdot$ Anz.Welten

Formelauswertung für Modallogik geht in polynomieller Rechenzeit.

Prozedur markiere(α):

falls $\alpha = \neg\beta, \Box\beta, \Diamond\beta$: markiere($\beta$)

falls $\alpha = \beta \wedge \gamma, \beta \vee \gamma, \beta \to \gamma$: markiere($\beta$); markiere($\gamma$)

für alle $w \in W$ (wir erweitern ggf. Formelmenge $\xi(w)$):

falls $\alpha = \neg\beta$ und $\beta \notin \xi[w]$: $\xi[w] = \xi[w] \cup \{\alpha\}$

falls $\alpha = \beta \wedge \gamma$ und $\beta \in \xi[w]$ und $\gamma \in \xi[w]$: $\xi[w] = \xi[w] \cup \{\alpha\}$

falls $\alpha = \beta \vee \gamma$, und $\beta \in \xi[w]$ oder $\gamma \in \xi[w]$: $\xi[w] = \xi[w] \cup \{\alpha\}$

falls $\alpha = \beta \to \gamma$, und $\beta \notin \xi[w]$ oder $\gamma \in \xi[w]$: $\xi[w] = \xi[w] \cup \{\alpha\}$

falls $\alpha = \Box\beta$, und für alle $(w, t) \in R$ gilt $\beta \in \xi(t)$: $\xi[w] = \xi[w] \cup \{\alpha\}$

falls $\alpha = \Diamond\beta$, und für ein $(w, t) \in R$ gilt $\beta \in \xi(t)$: $\xi[w] = \xi[w] \cup \{\alpha\}$

Hauptprogramm:

Eingabe $\mathcal{M}, s, \varphi$

($\mathcal{M} = (W, R, \xi)$ ist ein Kripke-Modell, $s \in W$, φ ist eine modallogische Formel)

(Für jedes $w \in W$ fassen wir $\xi[w]$ als globale Mengenvariable auf.)

für alle $w \in W$: $\xi[w] = \xi(w) \cup \{\top\}$.

markiere(φ)

Ausgabe $\varphi \in \xi[s]$

Gültige und erfüllbare modallogische Formeln

Definition 6.4 (gültig, erfüllbar)

Sei α eine modallogische Formel.

1. α heißt gültig (Schreibweise $\models_K \alpha$),
   wenn für jedes Kripke-Modell $\mathcal{M} = (W, R, \xi)$
   und jede Welt $w \in W$ gilt: $\mathcal{M}, w \models_K \alpha$.

2. α heißt erfüllbar,
   wenn es ein Kripke-Modell $\mathcal{M} = (W, R, \xi)$
   mit einer Welt $w \in W$ gibt, so dass $\mathcal{M}, w \models_K \alpha$.

Offensichtlich ist jede gültige/erfüllbare/unerfüllbare aussagenlogische Formel auch modallogisch gültig/erfüllbar/unerfüllbar.

Beispiel für eine gültige Formel

Die Gültigkeit modallogischer Formeln kann entsprechend der Gültigkeit aussagenlogischer Formeln bewiesen werden.

Bsp.: $\models_K \neg\Box\alpha \to \Diamond\neg\alpha$.

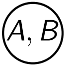
Beweis: Z.z: für alle $\mathcal{M}$ und w in $\mathcal{M}$ gilt $\mathcal{M}, w \models_K \neg\Box\alpha \to \Diamond\neg\alpha$.
Seien $\mathcal{M} = (W, R, \xi)$ und $w \in W$ beliebig.
Dann bleibt zu zeigen, dass $\mathcal{M}, w \models_K \neg\Box\alpha \to \Diamond\neg\alpha$ wahr ist.
Also formen wir die Aussage in eine offensichtlich wahre Aussage um.

$\mathcal{M}, w \models_K \neg\Box\alpha \to \Diamond\neg\alpha$ gdw.
$\mathcal{M}, w \not\models_K \neg\Box\alpha$ oder $\mathcal{M}, w \models_K \Diamond\neg\alpha$ gdw.
$\mathcal{M}, w \models_K \Box\alpha$ oder $\mathcal{M}, w \models_K \Diamond\neg\alpha$ gdw.
für alle $t \in W$: wenn $(w, t) \in R$, dann $\mathcal{M}, t \models_K \alpha$,
oder es gibt ein $t \in W$: $(w, t) \in R$ und $\mathcal{M}, t \models_K \neg\alpha$ gdw.
für alle $t \in W$: wenn $(w, t) \in R$, dann $\mathcal{M}, t \models_K \alpha$,
oder nicht für alle $t \in W$: wenn $(w, t) \in R$, dann $\mathcal{M}, t \models_K \alpha$ ✓

Erfüllbare Formeln (intuitiv)

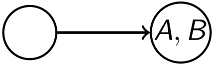
Um eine aussagenlogische Formel zu erfüllen,
braucht man nur eine Welt, deren Belegung die Formel erfüllt (Nachfolgewelten spielen keine Rolle).

$$A \wedge (B \to \neg C)$$



Um eine Formel $\Diamond\alpha$ zu erfüllen,
braucht man eine Welt mit einem Nachfolger, in dem α erfüllt wird.

$$\Diamond(A \wedge (B \to \neg C))$$



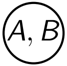
Um eine Formel $\Box\beta$ zu erfüllen,
reicht eine Welt ohne Nachfolger.
Falls die Welt bereits Nachfolger hat,
muss β in allen erfüllt werden.

$$\Box(A \to B)$$

Erfüllbare Formeln (intuitiv)

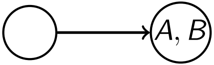
Um eine aussagenlogische Formel zu erfüllen,
braucht man nur eine Welt, deren Belegung die Formel erfüllt
(Nachfolgewelten spielen keine Rolle).

$$A \wedge (B \to \neg C)$$



Um eine Formel $\Diamond\alpha$ zu erfüllen,
braucht man eine Welt mit einem Nachfolger, in dem α erfüllt wird.

$$C \wedge \Diamond(A \wedge (B \to \neg C))$$



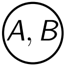
Um eine Formel $\Box\beta$ zu erfüllen,
reicht eine Welt ohne Nachfolger.
Falls die Welt bereits Nachfolger hat,
muss β in allen erfüllt werden.

$$\Box(A \to B)$$

Erfüllbare Formeln (intuitiv)

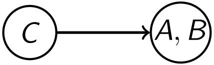
Um eine aussagenlogische Formel zu erfüllen,
braucht man nur eine Welt, deren Belegung die Formel erfüllt
(Nachfolgewelten spielen keine Rolle).

$$A \wedge (B \rightarrow \neg C)$$



Um eine Formel $\Diamond\alpha$ zu erfüllen,
braucht man eine Welt mit einem Nachfolger, in dem α erfüllt wird.

$$C \wedge \Diamond(A \wedge (B \rightarrow \neg C))$$



Um eine Formel $\Box\beta$ zu erfüllen,
reicht eine Welt ohne Nachfolger.
Falls die Welt bereits Nachfolger hat,
muss β in allen erfüllt werden.

$$\Box(A \rightarrow B)$$

Erfüllbare Formeln (intuitiv)

Um eine aussagenlogische Formel zu erfüllen,
braucht man nur eine Welt, deren Belegung die Formel erfüllt
(Nachfolgewelten spielen keine Rolle).

$$A \wedge (B \to \neg C) \qquad A, B$$

Um eine Formel $\Diamond\alpha$ zu erfüllen,
braucht man eine Welt mit einem Nachfolger, in dem α erfüllt wird.

$$C \wedge \Diamond(A \wedge (B \to \neg C)) \wedge \Diamond(\neg A \wedge B) \qquad C \longrightarrow A, B$$

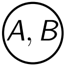
Um eine Formel $\Box\beta$ zu erfüllen,
reicht eine Welt ohne Nachfolger.
Falls die Welt bereits Nachfolger hat,
muss β in allen erfüllt werden.

$$\Box(A \to B) \qquad A$$

Erfüllbare Formeln (intuitiv)

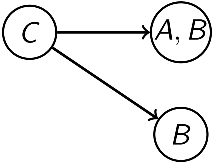
Um eine aussagenlogische Formel zu erfüllen,
braucht man nur eine Welt, deren Belegung die Formel erfüllt
(Nachfolgewelten spielen keine Rolle).

$$A \wedge (B \rightarrow \neg C)$$



Um eine Formel $\Diamond\alpha$ zu erfüllen,
braucht man eine Welt mit einem Nachfolger, in dem α erfüllt wird.

$$C \wedge \Diamond(A \wedge (B \rightarrow \neg C)) \wedge \Diamond(\neg A \wedge B)$$



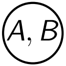
Um eine Formel $\Box\beta$ zu erfüllen,
reicht eine Welt ohne Nachfolger.
Falls die Welt bereits Nachfolger hat,
muss β in allen erfüllt werden.

$$\Box(A \rightarrow B)$$

Erfüllbare Formeln (intuitiv)

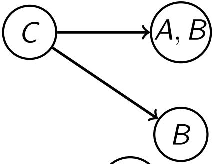
Um eine aussagenlogische Formel zu erfüllen,
braucht man nur eine Welt, deren Belegung die Formel erfüllt
(Nachfolgewelten spielen keine Rolle).

$$A \wedge (B \to \neg C)$$



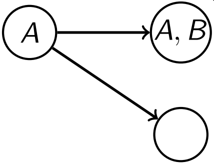
Um eine Formel $\Diamond\alpha$ zu erfüllen,
braucht man eine Welt mit einem Nachfolger, in dem α erfüllt wird.

$$C \wedge \Diamond(A \wedge (B \to \neg C)) \wedge \Diamond(\neg A \wedge B)$$



Um eine Formel $\Box\beta$ zu erfüllen,
reicht eine Welt ohne Nachfolger.
Falls die Welt bereits Nachfolger hat,
muss β in allen erfüllt werden.

$$\Box(A \to B) \wedge A \wedge \Diamond A \wedge \Diamond\neg A$$



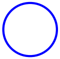
Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),
der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und
der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):
(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.
(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).

$$A \wedge \Diamond(B \wedge \Box G) \wedge \Diamond(G \wedge \neg B) \wedge \Box(\neg A \wedge \Diamond C \wedge \Diamond D \wedge \Box(\Diamond E \wedge \Diamond F))$$



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \land (\Diamond\delta_1 \land \ldots \land \Diamond\delta_k) \land \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),

der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und

der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \land (\Diamond\delta_1 \land \ldots \land \Diamond\delta_k) \land \Box\beta$ in Welt w):

(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.

(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \land \beta$ in w_i).

$$A \land \Diamond(B \land \Box G) \land \Diamond(G \land \neg B) \land \Box(\neg A \land \Diamond C \land \Diamond D \land \Box(\Diamond E \land \Diamond F))$$

Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),

der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und

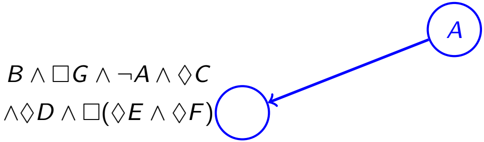
der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):

(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.

(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).

$$A \wedge \Diamond(B \wedge \Box G) \wedge \Diamond(G \wedge \neg B) \wedge \Box(\neg A \wedge \Diamond C \wedge \Diamond D \wedge \Box(\Diamond E \wedge \Diamond F))$$



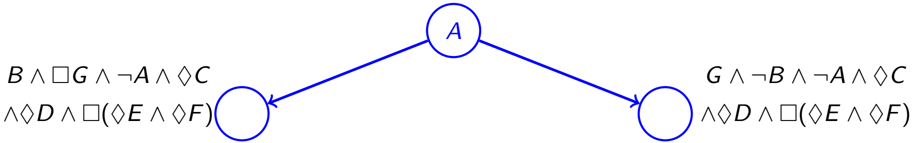
Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),
der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und
der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):
(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.
(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).

$$A \wedge \Diamond(B \wedge \Box G) \wedge \Diamond(G \wedge \neg B) \wedge \Box(\neg A \wedge \Diamond C \wedge \Diamond D \wedge \Box(\Diamond E \wedge \Diamond F))$$



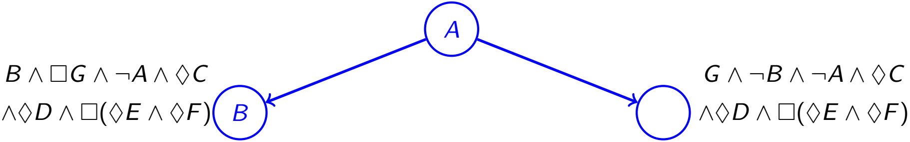
Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),
der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und
der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):
(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.
(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).

$$A \wedge \Diamond(B \wedge \Box G) \wedge \Diamond(G \wedge \neg B) \wedge \Box(\neg A \wedge \Diamond C \wedge \Diamond D \wedge \Box(\Diamond E \wedge \Diamond F))$$



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),

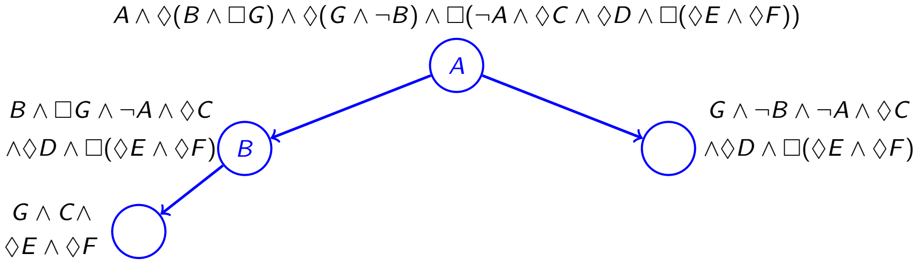
der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und

der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):

(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.

(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).



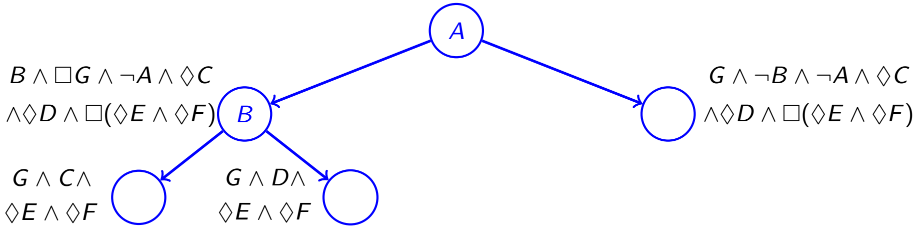
Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),
der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und
der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):
(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.
(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).

$$A \wedge \Diamond(B \wedge \Box G) \wedge \Diamond(G \wedge \neg B) \wedge \Box(\neg A \wedge \Diamond C \wedge \Diamond D \wedge \Box(\Diamond E \wedge \Diamond F))$$



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),

der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und

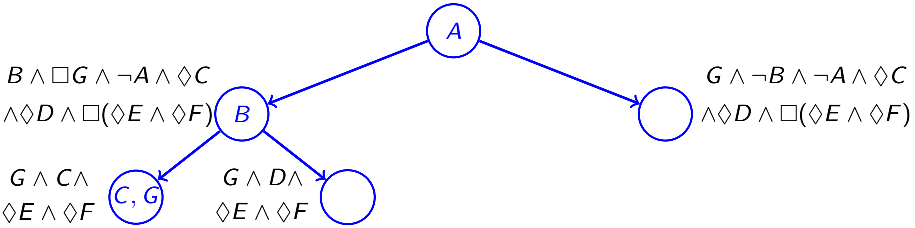
der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):

(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.

(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).

$$A \wedge \Diamond(B \wedge \Box G) \wedge \Diamond(G \wedge \neg B) \wedge \Box(\neg A \wedge \Diamond C \wedge \Diamond D \wedge \Box(\Diamond E \wedge \Diamond F))$$



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),

der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und

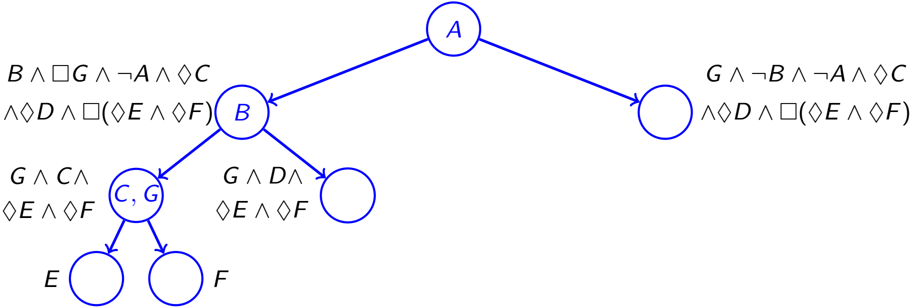
der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):

(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.

(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).

$$A \wedge \Diamond(B \wedge \Box G) \wedge \Diamond(G \wedge \neg B) \wedge \Box(\neg A \wedge \Diamond C \wedge \Diamond D \wedge \Box(\Diamond E \wedge \Diamond F))$$



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),

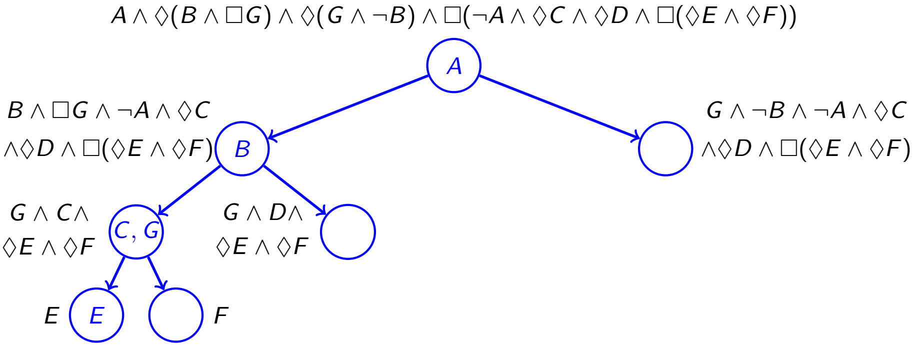
der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und

der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):

(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.

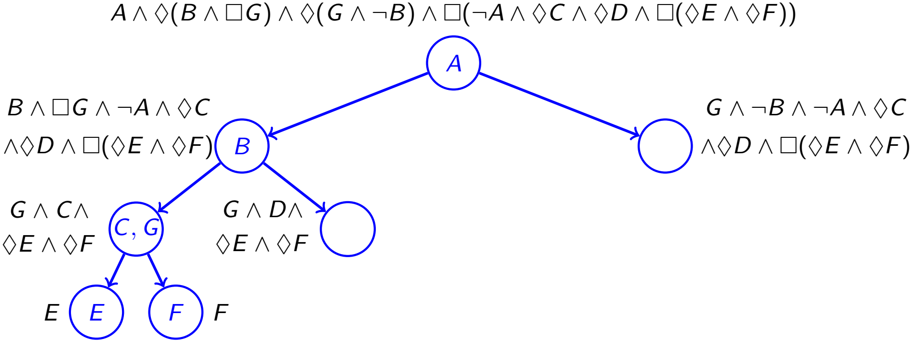
(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:
- der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),
- der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und
- der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):
(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.
(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),

der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und

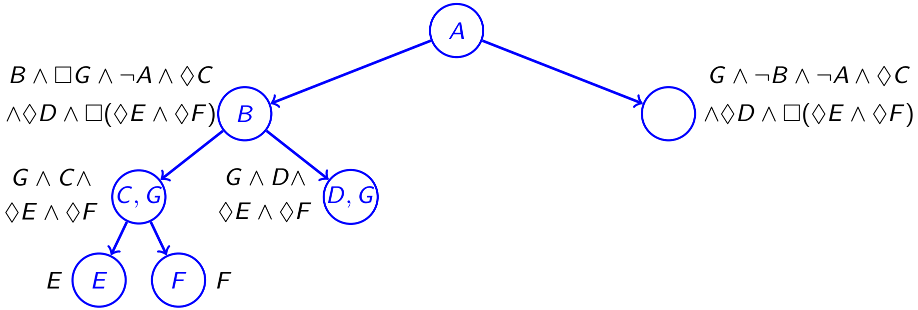
der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):

(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.

(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).

$$A \wedge \Diamond(B \wedge \Box G) \wedge \Diamond(G \wedge \neg B) \wedge \Box(\neg A \wedge \Diamond C \wedge \Diamond D \wedge \Box(\Diamond E \wedge \Diamond F))$$



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),

der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und

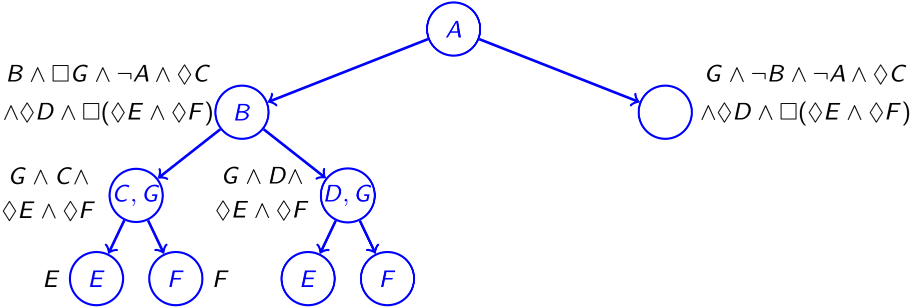
der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):

(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.

(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).

$$A \wedge \Diamond(B \wedge \Box G) \wedge \Diamond(G \wedge \neg B) \wedge \Box(\neg A \wedge \Diamond C \wedge \Diamond D \wedge \Box(\Diamond E \wedge \Diamond F))$$



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),

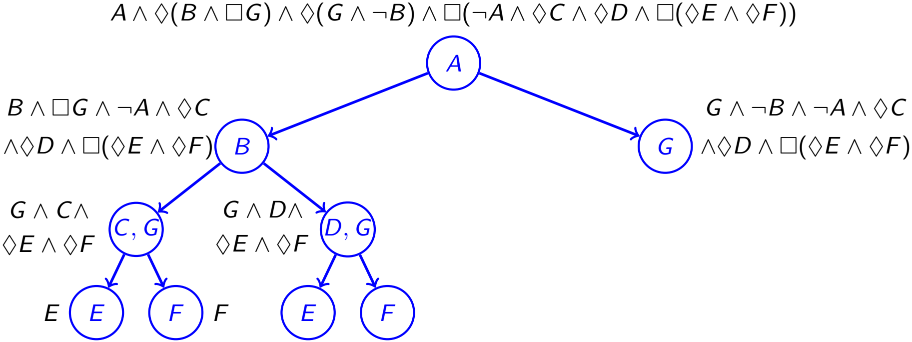
der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und

der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):

(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.

(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),

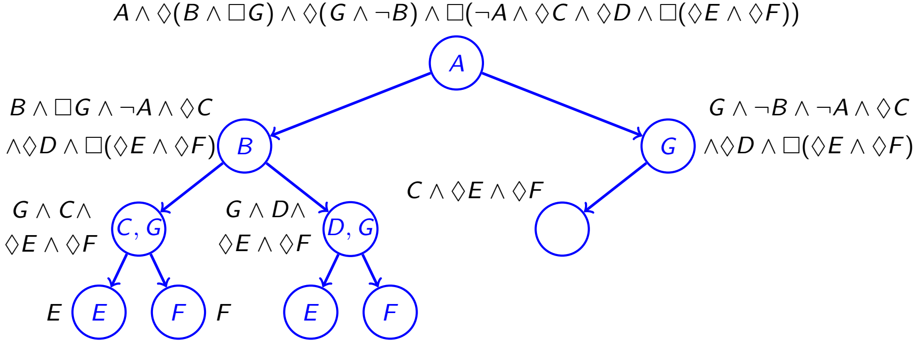
der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und

der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):

(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.

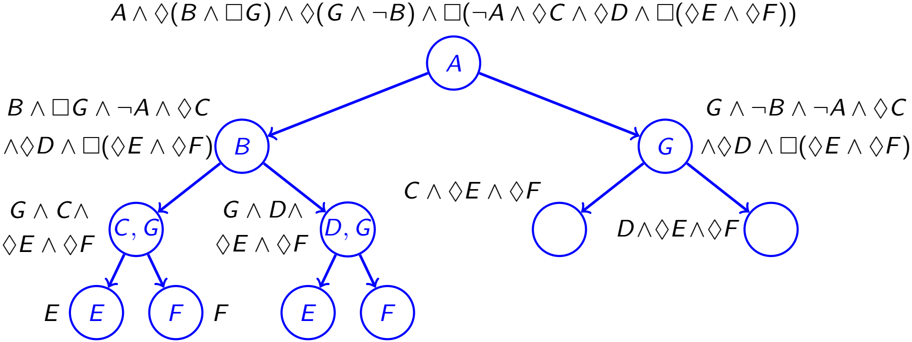
(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:
der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),
der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und
der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):
(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.
(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

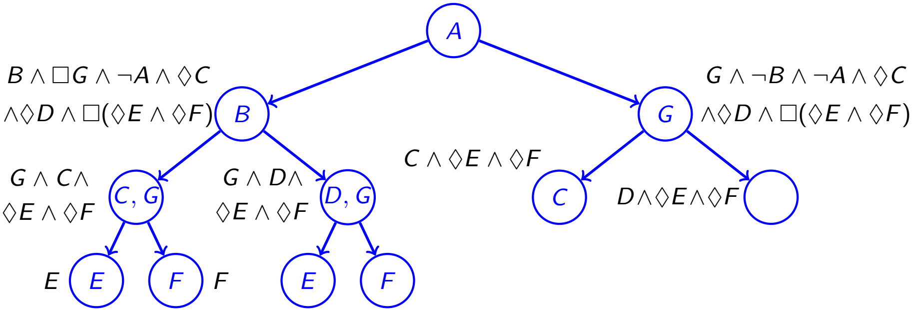
- der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),
- der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und
- der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):

(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.

(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).

$$A \wedge \Diamond(B \wedge \Box G) \wedge \Diamond(G \wedge \neg B) \wedge \Box(\neg A \wedge \Diamond C \wedge \Diamond D \wedge \Box(\Diamond E \wedge \Diamond F))$$



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),

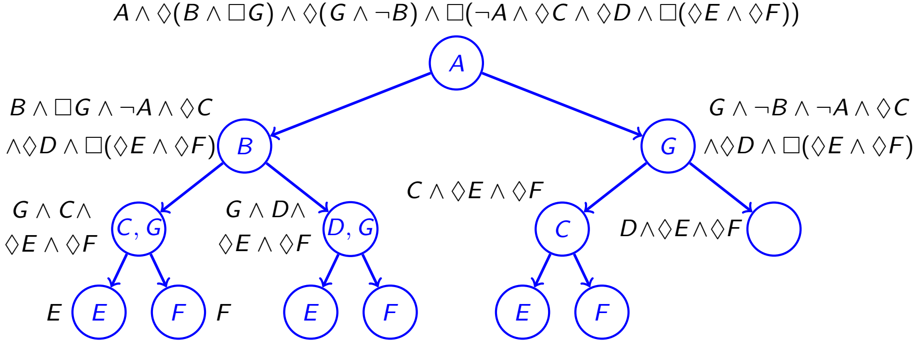
der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und

der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):

(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.

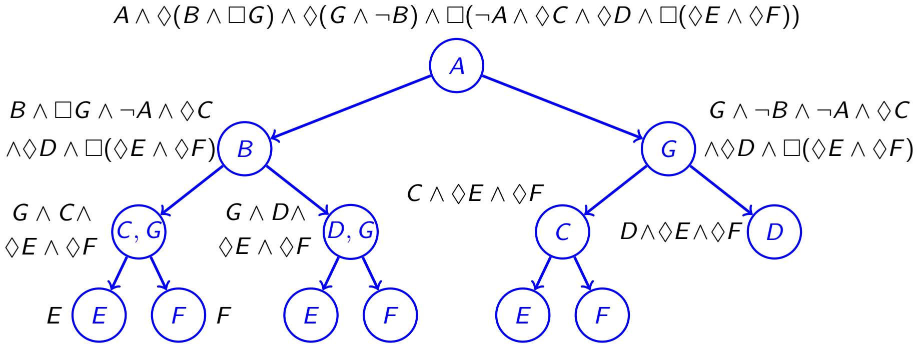
(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:
der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),
der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und
der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):
(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.
(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).



Konstruktion erfüllender Kripke-Modelle (grobe Idee, intuitiv)

Jede Formel hat drei Teile $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$:

der erste Teil α ist aussagenlogisch (ohne $\Box$ und $\Diamond$),

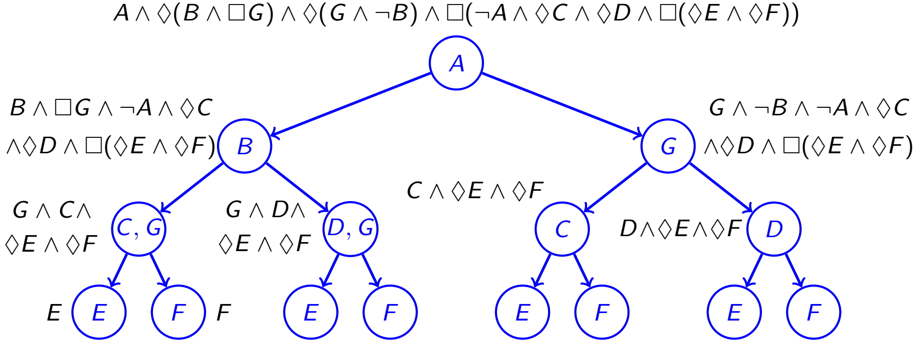
der zweite Teil besteht aus k Teilformeln, die mit $\Diamond$ beginnen, und

der dritte Teil ist eine Konjunktion aller Teilformeln, die mit $\Box$ beginnt.

Methode `erfülle`(Formel $\alpha \wedge (\Diamond\delta_1 \wedge \ldots \wedge \Diamond\delta_k) \wedge \Box\beta$ in Welt w):

(1) Erfülle α in Welt w durch Auswahl einer erfüllenden Belegung $\xi(w)$.

(2) Für $i = 1, \ldots, k$: erzeuge einen neuen Nachfolger w_i und `erfülle`($\delta_i \wedge \beta$ in w_i).



Äquivalente Formeln

Definition 6.5 (äquivalente Formeln)

Modallogische Formeln α und β heißen äquivalent ($\alpha \equiv \beta$), falls für jedes Kripke-Modell $\mathcal{M}$ und jede Welt w von $\mathcal{M}$ gilt:

$$\mathcal{M}, w \models_K \alpha \text{ gdw. } \mathcal{M}, w \models_K \beta.$$

Lemma 6.6 (wichtige Äquivalenzen)

Die Äquivalenzen für aussagenlogische Verknüpfungen (z.B. $\neg\neg\alpha \equiv \alpha$) gelten weiterhin. Außerdem gilt:

1. $\neg\Box\alpha \equiv \Diamond\neg\alpha$
2. $\neg\Diamond\alpha \equiv \Box\neg\alpha$
3. $\Box(\alpha \wedge \beta) \equiv \Box\alpha \wedge \Box\beta$
4. $\Diamond(\alpha \vee \beta) \equiv \Diamond\alpha \vee \Diamond\beta$
5. $\Diamond(\alpha \rightarrow \beta) \equiv \Box\alpha \rightarrow \Diamond\beta$

Beweis zu 2.:

Seien $\mathcal{M} = (W, R, \xi)$ und $w \in W$ beliebig gewählt.

Dann gilt:

$\mathcal{M}, w \models_K \neg\Diamond\alpha$

gdw. nicht $\mathcal{M}, w \models_K \Diamond\alpha$ (Semantik von $\neg$)

gdw. es gibt kein $t \in W$ mit $(w, t) \in R$ und $\mathcal{M}, t \models_K \alpha$ (Semantik von $\Diamond$)

gdw. für alle $t \in W$ gilt: wenn $(w, t) \in R$, dann $\mathcal{M}, t \not\models_K \alpha$ (ugs. Umformung)

gdw. für alle $t \in W$ gilt: wenn $(w, t) \in R$, dann $\mathcal{M}, t \models_K \neg\alpha$ (Semantik von $\neg$)

gdw. $\mathcal{M}, w \models_K \Box\neg\alpha$ (Semantik von $\Box$)

Die Beweise der übrigen Äquivalenzen gehen entsprechend. ✓

Adäquate Verküpfungszeichen

Lemma (2.7) kann für die Modallogik erweitert werden.

Lemma 6.7 (adäquate Verknüpfungszeichen für Modallogik)

Für jede modallogische Formel gibt es eine äquivalente Formel, die nur aus Atomen, $\bot$, $\rightarrow$ und $\Box$ besteht.

Beweis:
Der Beweis geht wie für Lemma (2.7).
Im Induktionsschritt müssen noch zusätzlich die Fälle $\varphi = \Box\alpha$ und $\varphi = \Diamond\alpha$ betrachtet werden.

Im Fall $\varphi = \Box\alpha$ folgt die äquivalente Formel $\Box\alpha'$ direkt aus der IV.

Im Fall $\varphi = \Diamond\alpha$ ergibt sich die äquivalente Formel wie folgt:

$$\begin{array}{lll} \Diamond\alpha & \equiv \neg\neg\Diamond\alpha & (\text{da } \beta \equiv \neg\neg\beta) \\ & \equiv \neg\Box\neg\alpha & (\text{da } \neg\Diamond\beta \equiv \Box\neg\beta \text{ (6.6)}) \\ & \equiv \neg\Box(\alpha \rightarrow \bot) & (\text{da } \neg\beta \equiv \beta \rightarrow \bot) \\ & \equiv \Box(\alpha \rightarrow \bot) \rightarrow \bot & (\text{da } \neg\beta \equiv \beta \rightarrow \bot) \\ & \equiv \Box(\alpha' \rightarrow \bot) \rightarrow \bot & (\text{IV}) \qquad \checkmark \end{array}$$

Definition 6.8 (Verallgemeinerungen von $\models_K$)

Sei α eine modallogische Formel und Γ eine Menge modallogischer Formeln.

1. Sei $\mathcal{M} = (W, R, \xi)$ ein Kripke-Modell.
   $\mathcal{M} \models_K \alpha$ bedeutet, dass $\mathcal{M}, w \models_K \alpha$ für jedes $w \in W$.
2. $\mathcal{M} \models_K \Gamma$ bedeutet, dass $\mathcal{M} \models_K \beta$ für jedes $\beta \in \Gamma$.
3. $\models_K \alpha$ bedeutet, dass $\mathcal{M} \models_K \alpha$ für jedes Kripke-Modell $\mathcal{M}$ („α ist gültig“).
4. Sei $G = (W, R)$ ein Graph.
   $G \models_K \alpha$ bedeutet, dass $\mathcal{M} \models_K \alpha$ für jede Belegung ξ und jedes Kripke-Modell $\mathcal{M} = (W, R, \xi)$.
5. $\Gamma \Vdash_K \alpha$ bedeutet, dass
   für jedes Kripke-Modell $\mathcal{M}$ gilt: wenn $\mathcal{M} \models_K \Gamma$, dann $\mathcal{M} \models_K \alpha$.

Was haben wir in Vorlesung 6 gelernt?

- Wir kennen modallogische Formeln mit $\Box$ und $\Diamond$.
- Wir wissen, was Kripke-Modelle sind und wie man sie graphisch darstellt.
- Wir kennen die Erfüllungsrelation für modallogische Formeln in Welten von Kripke-Modellen.
- Wir kennen einen Algorithmus zur Entscheidung der Erfüllungsrelation.
- Wir haben eine Vorstellung, wie man für eine erfüllbare Formel ein erfüllendes Kripke-Modelle konstruieren kann.
- Wir kennen den Äquivalenzbegriff für modallogische Formeln.
- Wir wissen, dass $\{\bot, \to, \Box\}$ adäquat für Modallogik ist.

Vorlesung 7:
Ein Tableau-Kalkül für Modallogik

Wir werden den modallogischen Tableau-Kalkül kennenlernen.
Er ist eine Erweiterung des aussagenlogischen Tableau-Kalküls um Expansionsregeln für $\Box$.

Wir gehen vor wie folgt.

- Definition von Tableaus für Formeln aus Atomen, $\bot$, $\rightarrow$ und $\Box$
- Definition von Tableau-Beweisen für Formeln
- Vollständigkeit des Tableau-Kalküls:
  wir zeigen, dass jede gültige Formel Tableau-beweisbar ist
- Korrektheit des Tableau-Kalküls:
  kommt später, nachdem wir Tableau-Beweise in Frege-Beweise umwandeln können

Der Begriff der semantischen Folgerung macht Tableaux „schwierig“.
Deswegen verzichten wir darauf und schauen uns nur Tableau-Beweise für die Gültigkeit von Formeln an
(wir betrachten also nur den Spezialfall $\models_{K} \alpha$ von $\Gamma \Vdash_{K} \alpha$).

7.1 Die Konstruktion eines Tableaus . . .

. . . entspricht in der Aussagenlogik einer systematischen Suche
nach einer erfüllenden Belegung der Startformel.

Entsprechend soll in der Modallogik nach einem Kripke-Modell gesucht werden, das die Startformel in einer Welt erfüllt.

Grundlegende Beobachtung: es reicht, „Baum-Modelle“ zu betrachten.

Tableau für $\Box A \to \Diamond A$

als einführendes Beispiel (stimmt nicht ganz genau …)

$$
\begin{tikzcd}
 & s, \Box A \to \Diamond A \arrow[dl] \arrow[dr] & \\
s, \neg\Box A \arrow[d] & & s, \Diamond A \arrow[d, "{(s, u)}"] \\
s, \Diamond\neg A \arrow[d, "{(s, t)}"] & & u, A \\
t, \neg A & &
\end{tikzcd}
$$

Tableau für $\Box A \to \Diamond A$

als einführendes Beispiel (stimmt nicht ganz genau ...)

$$
\begin{tikzcd}
 & s, \Box A \to \Diamond A \arrow[dl] \arrow[dr] & \\
s, \neg\Box A \arrow[d] & & s, \Diamond A \arrow[d, "{(s, u)}"] \\
s, \Diamond\neg A \arrow[d, "{(s, t)}"] & & u, A \\
t, \neg A & &
\end{tikzcd}
$$

Aus den nicht-wid.Pfaden ergeben sich folgende Modelle, die $\Box A \to \Diamond A$ in Welt s erfüllen:

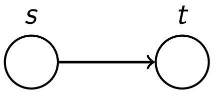
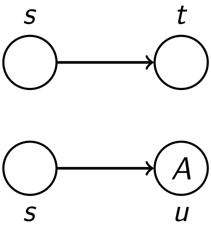


Tableau für $\Box A \to \Diamond A$

als einführendes Beispiel (stimmt nicht ganz genau ...)

$$
\begin{tikzcd}
 & s, \Box A \to \Diamond A \arrow[dl] \arrow[dr] & \\
s, \neg\Box A \arrow[d] & & s, \Diamond A \arrow[d, "{(s,u)}"] \\
s, \Diamond\neg A \arrow[d, "{(s,t)}"] & & u, A \\
t, \neg A & &
\end{tikzcd}
$$

Aus den nicht-wid.Pfaden ergeben sich folgende Modelle, die $\Box A \to \Diamond A$ in Welt s erfüllen:



7.2 Der Tableau-Kalkül für Modallogik

Der Tableau-Kalkül für Aussagenlogik wird erweitert.

Beim aussagenlogischen Tableau-Kalkül geht es darum,
für eine Formel einen Pfad durch das Tableau zu finden,
der eine erfüllende Belegung bestimmt.

Beim modallogischen Tableau-Kalkül soll ein Pfad π durch das Tableau gefunden werden,
der ein erfüllendes Kripke-Modell (W_π, R_π, ξ_π) mit
einer Welt s bestimmt, in der die Formel erfüllt wird.

Dazu werden die Tableaux erweitert.

1. Markierungen der Knoten haben die Form w, α mit Welt w und Formel α. W_π ist die Menge aller Welten, die auf Pfad π vorkommen. Der Startknoten des Tableaus für α wird mit s, α markiert.
2. Die Kanten des Tableaus werden teilweise mit Kanten zwischen Welten markiert. R_π ist die Menge aller Kantenmarkierungen auf dem Pfad π.

Definition 7.1 (Tableau für modallogische Formeln aus Atomen, $\bot$, $\to$ und $\Box$)

Sei φ eine modallogische Formel aus Atomen, $\bot$, $\to$ und $\Box$.

1. Ein Knoten, der mit s, φ ist, ist ein Tableau für φ.
2. Sei T ein Tableau für φ, und κ sei ein Knoten in T mit Markierung w, ψ.
   Wenn ψ eine Formel der Form $\alpha \to \beta$ (mit $\beta \neq \bot$) oder $\neg(\alpha \to \beta)$ ist, dann kann κ mit der entsprechenden Expansionsregel expandiert werden.

   $w, \alpha \to \beta$: $\begin{array}{ccc} & \bullet & \\ \swarrow & & \searrow \\ w, \neg\alpha & & w, \beta \end{array}$ $\qquad$ $w, \neg(\alpha \to \beta)$: $\begin{array}{c} \bullet \\ \downarrow \\ w, \alpha \\ \downarrow \\ w, \neg\beta \end{array}$

   Wenn ψ eine Formel der Form $\neg\Box\alpha$ ist und kein Pfad durch κ einen Knoten hat, der nach den obigen beiden Regeln expandiert werden kann,
   dann kann κ mit folgenden Expansionsregeln expandiert werden. Dabei ist z eine „neue“ Welt, die bisher in keiner Markierung des Tableaus vorkommt.

   $w, \neg\Box\alpha$: $\begin{array}{c} \bullet \\ \Big\downarrow (w, z) \\ z, \neg\alpha \end{array}$ $\qquad$ für jedes $w, \Box\beta$ auf jedem Pfad mit diesem z: $\begin{array}{c} \bullet \\ \Big\downarrow \\ z, \beta \end{array}$

   Durch Expansion von κ entsteht ein (weiteres) Tableau für φ.

Beispiel: Tableau für $\neg(\Box A \to \Diamond A)$

$$
\begin{array}{c}
s, \neg(\Box A \to \neg\Box\neg A) \\
\downarrow \\
s, \Box A \\
\downarrow \\
s, \neg\neg\Box\neg A \\
\downarrow \\
s, \Box\neg A \\
\downarrow \\
s, \neg\bot
\end{array}
$$

Wir benutzen $\Diamond\alpha$ als abkürzende Schreibweise für $\neg\Box\neg\alpha$.

$s, \Box A$ wird nur zusammen mit $s, \neg\Box\ldots$ expandiert.

Bei der Expansion von $s, \neg(\underbrace{\Box\neg A \to \bot}_{\neg\Box\neg A})$ entstehen Knoten mit den Markierungen $s, \Box\neg A$ und $s, \neg\bot$.

Knoten mit Markierung $x, \neg\bot$ werden wir in Zukunft weglassen, da sie keinen Beitrag zu widersprüchlichen Pfaden oder Belegungen leisten.

Beispiel: Tableau für $\neg(\Box A \to \Diamond A)$

$$\begin{array}{c} s, \neg(\Box A \to \neg\Box\neg A) \\ \downarrow \\ s, \Box A \\ \downarrow \\ s, \neg\neg\Box\neg A \\ \downarrow \\ s, \Box\neg A \\ \downarrow \\ s, \neg\bot \end{array}$$

Wir benutzen $\Diamond\alpha$ als abkürzende Schreibweise für $\neg\Box\neg\alpha$.

$s, \Box A$ wird nur zusammen mit $s, \neg\Box\ldots$ expandiert.

Bei der Expansion von $s, \neg(\underbrace{\Box\neg A \to \bot}_{\neg\Box\neg A})$ entstehen

Knoten mit den Markierungen $s, \Box\neg A$ und $s, \neg\bot$.

Knoten mit Markierung $x, \neg\bot$
werden wir in Zukunft weglassen, da sie keinen Beitrag zu widersprüchlichen Pfaden oder Belegungen leisten.

Beispiel: Tableau für $\neg(\Box A \to \Diamond A)$

$$\begin{array}{c} s, \neg(\Box A \to \neg\Box\neg A) \\ \downarrow \\ s, \Box A \\ \downarrow \\ s, \neg\neg\Box\neg A \\ \downarrow \\ s, \Box\neg A \\ \downarrow \\ s, \neg\bot \end{array}$$

Wir benutzen $\Diamond\alpha$ als abkürzende Schreibweise für $\neg\Box\neg\alpha$.

$s, \Box A$ wird nur zusammen mit $s, \neg\Box\ldots$ expandiert.

Bei der Expansion von $s, \neg(\underbrace{\Box\neg A \to \bot}_{\neg\Box\neg A})$ entstehen Knoten mit den Markierungen $s, \Box\neg A$ und $s, \neg\bot$.

Knoten mit Markierung $x, \neg\bot$
werden wir in Zukunft weglassen, da sie keinen Beitrag zu widersprüchlichen Pfaden oder Belegungen leisten.

Beispiel: Tableau für $\neg(\Box A \rightarrow \Diamond A)$

$$\begin{array}{c} s, \neg(\Box A \rightarrow \neg\Box\neg A) \\ \downarrow \\ s, \Box A \\ \downarrow \\ s, \neg\neg\Box\neg A \\ \downarrow \\ s, \Box\neg A \\ \downarrow \\ s, \neg\bot \end{array}$$

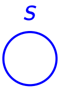
Wir benutzen $\Diamond\alpha$ als abkürzende Schreibweise für $\neg\Box\neg\alpha$.

$s, \Box A$ wird nur zusammen mit $s, \neg\Box\ldots$ expandiert.

Bei der Expansion von $s, \neg(\underbrace{\Box\neg A \rightarrow \bot}_{\neg\Box\neg A})$ entstehen Knoten mit den Markierungen $s, \Box\neg A$ und $s, \neg\bot$.

Knoten mit Markierung $x, \neg\bot$
werden wir in Zukunft weglassen, da sie keinen Beitrag zu widersprüchlichen Pfaden oder Belegungen leisten.

Erfüllendes Modell, das durch den Pfad bestimmt wird:

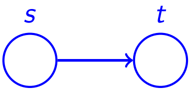


Beispiel: Tableau für

$$s, \neg(\Diamond(A \to B) \to (\Box A \to \Diamond B))$$

Wir benutzen $\Diamond\alpha$
als Abkürzung für $\neg\Box\neg\alpha$.

Graph, der durch jeden der beiden Pfade bestimmt wird:



$$\begin{array}{c}
s, \neg(\Diamond(A \to B) \to (\Box A \to \Diamond B)) \\
\downarrow \\
s, \neg\Box\neg(A \to B) \\
\downarrow \\
s, \neg(\Box A \to \neg\Box\neg B) \\
\downarrow \\
s, \Box A \\
\downarrow \\
s, \neg\neg\Box\neg B \\
\downarrow \\
s, \Box\neg B \\
\downarrow \\
s, \Box\neg B \\
\downarrow (s, t) \\
t, \neg\neg(A \to B) \\
\downarrow \\
t, A \to B \\
\downarrow \\
t, A \\
\downarrow \\
t, \neg B \\
\downarrow \\
t, A \to B \\
\swarrow \qquad \searrow \\
t, \neg A \ \times \qquad\qquad t, B \ \times
\end{array}$$

Definition 7.2 (Eigenschaften von Pfaden und Tableaux)

Ein Pfad durch ein Tableau heißt widersprüchlich, wenn
er eine Markierung $t, \perp$ oder zwei widersprüchliche Markierungen
t, α und $t, \neg\alpha$ für eine Welt t und eine Formel α enthält.

Vereinbarung: sobald ein Pfad widersprüchlich ist, wird kein weiterer Knoten auf ihm expandiert.

Ein Pfad durch ein Tableau heißt geschlossen, wenn
er nicht widersprüchlich ist und
jeder (expandierbare) Knoten auf ihm expandiert wurde.

Ein Tableau heißt geschlossen,
wenn alle Pfade durch das Tableau geschlossen oder widersprüchlich sind.

Ein Tableau heißt widersprüchlich,
wenn alle Pfade durch das Tableau widersprüchlich sind.

$$
\begin{tikzcd}[column sep=small]
 & s, \neg(\neg\Box A \to (\neg\Box B \to \neg\Box(\neg B \to A))) \arrow[d] & & & \\
 & s, \neg\Box A \arrow[d] & & & \\
 & s, \neg(\neg\Box B \to \neg\Box(\neg B \to A)) \arrow[d] & & & \\
 & s, \neg\Box B \arrow[d] & & & \\
 & s, \neg\neg\Box(\neg B \to A) \arrow[d] & & & \\
 & s, \Box(\neg B \to A) \arrow[d, "{(s,t) \in R}"] & & & \\
 & t, \neg A \arrow[d] & & & \\
 & t, \neg B \to A \arrow[dl] \arrow[dr] & & & \\
t, \neg\neg B \arrow[rrr, bend right] & & t, A & t, B \arrow[d, "{(s,u) \in R}"] & \\
 & & \times & u, \neg B \arrow[d] & \\
 & & & u, \neg B \to A \arrow[dl] \arrow[dr] & \\
 & & u, \neg\neg B & & u, A \\
 & & \times & &
\end{tikzcd}
$$

$s, \neg(\neg\Box A \to (\neg\Box B \to \neg\Box(\neg B \to A)))$

$\downarrow$

$s, \neg\Box A$

$\downarrow$

$s, \neg(\neg\Box B \to \neg\Box(\neg B \to A))$

$\downarrow$

$s, \neg\Box B$

$\downarrow$

$s, \neg\neg\Box(\neg B \to A)$

$\downarrow$

$s, \Box(\neg B \to A)$

$\downarrow (s, t) \in R$

$t, \neg A$

$\downarrow$

$t, \neg B \to A$

$\swarrow \qquad \searrow$

$t, \neg\neg B \qquad t, A$

$\qquad\qquad\quad \times$

t, B

$\downarrow (s, u) \in R$

$u, \neg B$

$\downarrow$

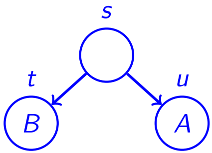
$u, \neg B \to A$

$\swarrow \qquad \searrow$

$u, \neg\neg B \qquad u, A$

$\times$

Modell (W_π, R_π, ξ_π), das durch den blauen Pfad π bestimmt wird:



Es gibt modallogische Formeln der Länge n,
die nur von Baum-Modellen mit 2^n Welten erfüllt werden.
Deshalb können in modalen Tableaux die Pfade sehr lang werden.

Damit erhalten wir auch eine schlechtere Schranke für die Größe modaler Tableaux als im aussagenlogischen Fall.

Wichtig ist jedoch, dass man systematisch endliche Tableaux erzeugen kann.

Lemma 7.3 (endliche Tableaux reichen)

Für jede modallogische Formel α gibt es ein geschlossenes Tableau mit $\leqslant 2^{2^{|\alpha|}}$ Pfaden.

Definition 7.4 (Tableau-beweisbar)

Sei α eine modallogische Formel.

Die Relation $\vdash_{\Box\mathsf{Tab}}$ für modallogische Formeln ist definiert durch:

$\vdash_{\Box\mathsf{Tab}} \alpha$ gdw. es ein widersprüchliches Tableau für $\neg\alpha$ gibt.

Beispiel: $\vdash_{\Box\mathrm{Tab}} \Box(A \to B) \to (\Box A \to \Box B)$

$$
\begin{tikzcd}[row sep=small]
 & s, \neg(\Box(A \to B) \to (\Box A \to \Box B)) \arrow[d] & \\
 & s, \Box(A \to B) \arrow[d] & \\
 & s, \neg(\Box A \to \Box B) \arrow[d] & \\
 & s, \Box A \arrow[d] & \\
 & s, \neg\Box B \arrow[d, "{(s,t)}"] & \\
 & t, \neg B \arrow[d] & \\
 & t, A \to B \arrow[d] & \\
 & t, A \arrow[dl] \arrow[dr] & \\
t, \neg A & & t, B \\
\times & & \times
\end{tikzcd}
$$

7.3 Der Beweis des Vollständigkeitslemmas . . .

. . . ist eine Erweiterung des Beweises für aussagenlogische Formeln.

Dort hatten wir gezeigt:

jeder geschlossene Pfad durch ein Tableau für φ
bestimmt eine Belegung, die φ erfüllt.

Hier werden wir zeigen, dass

jeder geschlossene Pfad durch ein Tableau für φ
ein Kripke-Modell mit einer Welt bestimmt, die φ erfüllt.

Lemma 7.5 (Pfad bestimmt Modell)

Sei T ein modales Tableau für eine Formel φ, und π ein nicht-widersprüchlicher Pfad durch T, auf dem jeder Knoten expandiert ist.
Dann gilt für das Kripke-Modell $\mathcal{M}_\pi = (W_\pi, R_\pi, \xi_\pi)$
mit $\xi_\pi(w) = \{A_i \mid \; w, A_i \text{ kommt auf } \pi \text{ vor}\}$:
für jede Markierung v, α auf π gilt $\mathcal{M}_\pi, v \models_K \alpha$.

Beweis: sei π ein solcher Pfad mit der Knotenfolge $\pi_1, \ldots, \pi_\ell$.
Wir zeigen induktiv, dass die Behauptung für jedes π_i gilt.
IA: $i = \ell$. (D.h. wir betrachten den Knoten π_ℓ am Ende von π.)
Fall 1: π_ℓ hat Markierung $v, \neg\bot$, v, A_j oder $v, \neg A_j$ für ein Atom A_j.
Es gilt $\mathcal{M}_\pi, v \models_K \neg\bot$, und $\mathcal{M}_\pi, v \models_K A_j$ bzw. $\mathcal{M}_\pi, v \models_K \neg A_j$
aufgrund der Definition von ξ_π und der Nicht-Widersprüchlichkeit von π.
Fall 2: π_ℓ ist ein Knoten mit Markierung $v, \Box\beta$.
Da π_ℓ nicht expandiert werden musste, hat v keinen Nachfolger in R_π.
Also gilt $\mathcal{M}, v \models_K \Box\beta$.

Eine andere Markierung kann π_ℓ nicht haben.

IV: Für $\pi_k, \pi_{k+1}, \ldots, \pi_\ell$ mit $\pi_i = v_i, \alpha_i$ gilt $\mathcal{M}_\pi, v_i \models_K \alpha_i$.

IS: π_{k-1} habe Markierung v, α. Zu zeigen ist: $\mathcal{M}_\pi, v \models_K \alpha$.

Fall 1: v, α wird nicht expandiert, d.h.

$\alpha = \neg\bot$, $\alpha = A_j$ oder $\alpha = \neg A_j$ für ein Atom A_j,

oder $\alpha = \Box\beta$ und v hat keinen Nachfolger in R_π.

Dann geht die Argumentation wie im IA.

Fall 2: $\alpha = (\beta \to \gamma)$ (mit $\gamma \neq \bot$) oder $\alpha = \neg(\beta \to \gamma)$.

Dann wird α „klassisch“ aussagenlogisch expandiert und die Argumentation ist wie im aussagenlogischen Fall (Beweis zu 5.3).

Fall 3: $\alpha = \neg\Box\beta$.

Dann wird $\pi_{k-1}(= v, \alpha)$ expandiert.
Also gibt es ein π_j mit Markierung $t, \neg\beta$,
so dass $j > k-1$ und $(v, t) \in R_\pi$.
Nach IV gilt $\mathcal{M}_\pi, t \models_K \neg\beta$, d.h. $\mathcal{M}_\pi, t \not\models_K \beta$.
Also gilt nicht für alle $u \in W_\pi$ mit $(v, u) \in R_\pi$, dass $\mathcal{M}_\pi, u \models_K \beta$,
und daraus folgt $\mathcal{M}_\pi, v \models_K \neg\Box\beta$, d.h. $\mathcal{M}_\pi, v \models_K \alpha$.

Fall 4: $\alpha = \Box\beta$ und $\pi_{k-1}(= v, \alpha)$ wird expandiert.

Für jede Kante $(v, t) \in R_\pi$ gilt:
Es gibt ein π_j mit $j > k-1$ und Markierung t, β.
Nach IV gilt dann für jede Kante $(v, t) \in R_\pi$: $\quad \mathcal{M}_\pi, t \models_K \beta$.
Also folgt $\mathcal{M}_\pi, v \models_K \Box\beta$, d.h. $\mathcal{M}_\pi, v \models_K \alpha$. ✓

Lemma 7.6 (Vollständigkeit des modalen Tableau-Kalküls)

Sei α eine modallogische Formel.
Aus $\models_K \alpha$ folgt $\vdash_{\Box\mathsf{Tab}} \alpha$.

Beweis:
Wir beweisen die Kontraposition.

Gelte $\not\vdash_{\Box\mathsf{Tab}} \alpha$.
Dann besitzt $\neg\alpha$ ein endliches nicht-widersprüchliches geschlossenes Tableau T_0 (7.3), das mit $s, \neg\alpha$ beginnt.
Also gibt es einen geschlossenen Pfad π durch T_0.
Nach Lemma (7.5) gilt $\mathcal{M}_\pi, s \models_K \neg\alpha$.
Also folgt $\not\models_K \alpha$. ✓

Was haben wir in Vorlesung 7 gelernt?

Wir haben den modallogischen Tableau-Kalkül kennengelernt.

- Wir können geschlossene Tableaux für modallogische Formeln konstruieren.
- Wir kennen $\vdash_{\Box\mathsf{Tab}}$ und den Begriff der Tableau-Beweisbarkeit.
- Wir wissen, wie man aus einem geschlossenen Pfad durch ein Tableau ein Kripke-Modell erhält, das alle Markierungen auf dem Pfad „erfüllt“. Wir können den Beweis mittels Induktion über die Pfadlänge reproduzieren.
- Wir können damit das Vollständigkeitslemma für den Tableau-Kalkül beweisen.

Vorlesung 8:
Ein Frege-Kalkül für Modallogik

Nun werden wir einen modallogischen Frege-Kalkül kennenlernen.
Wir gehen vor wie folgt.

- Definition von Axiomen und Schlussregeln für Frege-Beweise – das ist eine Erweiterung des al. Frege-Kalküls um ein Axiom und eine Schlussregel
- Formulierung eines modifizierten Deduktionstheorems
- Korrektheit des Frege-Kalküls:
  dazu erweitern wir den Korrektheitsbeweis des al. Frege-Kalküls für das neue Axiom und die neue Schlussregel
- Vollständigkeit des Frege-Kalküls:
  wir benutzen die Vollständigkeit des Tableau-Kalküls und zeigen, wie man Tableau-Beweise in Frege-Beweise umwandeln kann – dazu erweitern wir das Vorgehen in der Aussagenlogik (innerhalb einer Welt) um den Übergang zwischen Welten

8.1 Ein Frege-Kalkül für Modallogik

Definition 8.1 ($\Box$Frege-Kalkül und $\vdash_{\Box\text{Fre}}$)

1. Die *Elemente* des $\Box$Frege-Kalküls sind
modallogische Formeln aus Atomen, $\bot$, $\rightarrow$ und $\Box$.

2. Die Axiome des $\Box$Frege-Kalküls sind für alle Formeln α, β, φ:

(A1) $\alpha \rightarrow (\beta \rightarrow \alpha)$

(A2) $(\alpha \rightarrow (\beta \rightarrow \varphi)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \varphi))$

(A3) $\neg\neg\alpha \rightarrow \alpha$

(K) $\Box(\alpha \rightarrow \beta) \;\rightarrow\; (\Box\alpha \rightarrow \Box\beta)$

3. Die Schlussregeln des $\Box$Frege-Kalküls sind
modus ponens (MP): aus α und $\alpha \rightarrow \beta$ kann man β herleiten, und
Generalisierung (Gen): aus α kann man $\Box\alpha$ herleiten.

$$\frac{\alpha \qquad \alpha \rightarrow \beta}{\beta} \qquad\qquad \frac{\alpha}{\Box\alpha}$$

(Fortsetzung von Definition 8.1)

4. Eine Herleitung einer Formel α im $\Box$Frege-Kalkül ist eine Folge $\alpha_1, \alpha_2, \ldots, \alpha_\ell$ von Formeln, an deren Ende $\alpha (= \alpha_\ell)$ steht und deren Elemente folgende Eigenschaften haben (für $i = 1, 2, \ldots, \ell$):

- α_i ist ein Axiom, oder
- es gibt α_a mit $a < i$, aus dem α_i in einem Schritt mittels Generalisierung hergeleitet werden kann (d.h. es gibt $a < i$ mit $\alpha_i = \Box\alpha_a$), oder
- es gibt α_a, α_b mit $a, b < i$, aus denen α_i in einem Schritt mit modus ponens hergeleitet werden kann (d.h. es gibt $a, b < i$ mit $\alpha_b = \alpha_a \to \alpha_i$).

(Statt *Herleitung* verwendet man gerne auch (Frege-)Beweis.)

5. $\vdash_{\Box\mathsf{Fre}} \alpha$, falls es eine Herleitung der Formel α im $\Box$Frege-Kalkül gibt.

Lemma 8.2 (Doppelnegation überspringt modale Operatoren)

Für alle modallogischen Formeln α gilt

1. $\vdash_{\overline{\Box\mathsf{Fre}}} \neg\neg\Box\alpha \to \Box\neg\neg\alpha$ *und*

2. $\vdash_{\overline{\Box\mathsf{Fre}}} \Box\neg\neg\alpha \to \neg\neg\Box\alpha$.

Beweis:

(1)	$\alpha \to \neg\neg\alpha$	(4.13(2))
(2)	$\Box(\alpha \to \neg\neg\alpha)$	Gen (1)
(3)	$\Box(\alpha \to \neg\neg\alpha) \to (\Box\alpha \to \Box\neg\neg\alpha)$	(K)
(4)	$\Box\alpha \to \Box\neg\neg\alpha$	MP (2), (3)
(5)	$\neg\neg\Box\alpha \to \Box\alpha$	(A3)
(6)	$\neg\neg\Box\alpha \to \Box\neg\neg\alpha$	TRANS (5), (4)

(Teil 2 ist Übungsaufgabe.) ✓

Das Deduktionstheorem von $\vdash_{\text{Fre}}$ geht nicht bei $\vdash_{\Box\text{Fre}}$

(1)		$\vdash_{\Box\text{Fre}}$	$\alpha \to \alpha$	(geht wie bei $\vdash_{\text{Fre}}$)
(2)	α	$\vdash_{\Box\text{Fre}}$	α	(Anwendung des DT „von rechts nach links“ geht wie bei $\vdash_{\text{Fre}}$
(3)	α	$\vdash_{\Box\text{Fre}}$	$\Box\alpha$	(Gen liefert einen nicht-korrekten Sequenten, da $\alpha \not\models \Box\alpha$)
(4)		$\vdash_{\Box\text{Fre}}$	$\alpha \to \Box\alpha$	(Ist nicht korrekt, da z.B. $A \to \Box A$ nicht gültig ist.)

Man kann bei Beweisen in $\vdash_{\Box\text{Fre}}$ nur dann „Hypothesen von links nach rechts verschieben“, wenn man Gen nur auf Sequenten ohne Hypothesen anwendet.

Der Rest vom Deduktionstheorem

Satz 8.3 (Deduktionstheorem für $\vdash_{\Box\mathsf{Fre}}$)

Sei Γ eine Menge modallogischer Formeln, α und β seien modallogische Formeln.

1. *Falls $\Gamma \vdash_{\Box\mathsf{Fre}} \alpha \to \beta$, dann auch $\Gamma \cup \{\alpha\} \vdash_{\Box\mathsf{Fre}} \beta$.*
2. *Falls $\Gamma \cup \{\alpha\} \vdash_{\Box\mathsf{Fre}} \beta$ mittels einer Herleitung, bei der die Schlussregel Gen nur auf Formeln angewendet wird, die ohne Hypothesen herleitbar sind, dann $\Gamma \vdash_{\Box\mathsf{Fre}} \alpha \to \beta$.*

Der Beweis geht wie im aussagenlogischen Fall (siehe (4.6)).

Beim vereinfachten Aufschreiben von Herleitungen mit Hypothesen, bei der man DT verwenden will, muss man diese eingeschränkte Benutzung von Gen beachten.
Dann kann man (wie gewohnt) Hypothesen benutzen.

Lemma 8.4 (Verallgemeinerung von (K))

Für jedes $k \geqslant 1$ und alle Formeln $\alpha_1, \ldots, \alpha_k, \beta$ gilt

$$\vdash_{\Box\mathrm{Fre}} \Box(\alpha_1 \to (\alpha_2 \to (\alpha_3 \to \ldots (\alpha_k \to \beta) \ldots))) \to (\Box\alpha_1 \to (\Box\alpha_2 \to (\Box\alpha_3 \to \ldots (\Box\alpha_k \to \Box\beta) \ldots)))$$

Beweis mittels Induktion über k.

IA $k = 1$: für alle α_1, β ist $\Box(\alpha_1 \to \beta) \to (\Box\alpha_1 \to \Box\beta)$ Axiom (K). ✓

IV: für beliebiges festes k und alle Formeln $\alpha_1, \ldots, \alpha_k, \beta$ gilt

$$\vdash_{\Box\mathrm{Fre}} \Box(\alpha_1 \to (\alpha_2 \to (\alpha_3 \to \ldots (\alpha_k \to \beta) \ldots))) \to (\Box\alpha_1 \to (\Box\alpha_2 \to (\Box\alpha_3 \to \ldots (\Box\alpha_k \to \Box\beta) \ldots))).$$

IS:

(1) $\vdash_{\Box\mathrm{Fre}} \Box(\alpha_1 \to (\alpha_2 \to (\alpha_3 \to \ldots(\alpha_{k+1} \to \beta)\ldots))) \to$
$(\Box\alpha_1 \to \Box(\alpha_2 \to (\alpha_3 \to \ldots(\alpha_{k+1} \to \beta)\ldots)))$ (K)

(2) $\Box(\alpha_1 \to (\alpha_2 \to (\alpha_3 \to \ldots(\alpha_{k+1} \to \beta)\ldots))), \Box\alpha_1 \vdash_{\Box\mathrm{Fre}}$
$\Box(\alpha_2 \to (\alpha_3 \to \ldots(\alpha_{k+1} \to \beta)\ldots))$ DT (1)

(3) $\vdash_{\Box\mathrm{Fre}} \Box(\alpha_2 \to (\alpha_3 \to (\alpha_4 \to \ldots(\alpha_{k+1} \to \beta)\ldots))) \to$
$(\Box\alpha_2 \to (\Box\alpha_3 \to (\Box\alpha_4 \to \ldots(\Box\alpha_{k+1} \to \Box\beta)\ldots)))$ IV

(4) $\Box(\alpha_1 \to (\alpha_2 \to (\alpha_3 \to \ldots(\alpha_{k+1} \to \beta)\ldots))), \Box\alpha_1 \vdash_{\Box\mathrm{Fre}}$
$\Box\alpha_2 \to (\Box\alpha_3 \to \ldots(\Box\alpha_{k+1} \to \Box\beta)\ldots)$ MP (2),(3)

(5) $\vdash_{\Box\mathrm{Fre}} \Box(\alpha_1 \to (\alpha_2 \to (\alpha_3 \to \ldots(\alpha_{k+1} \to \beta)\ldots))) \to$
$(\Box\alpha_1 \to (\Box\alpha_2 \to (\Box\alpha_3 \to \ldots(\Box\alpha_{k+1} \to \Box\beta)\ldots)))$ DT (4)

Da Gen nicht benutzt wurde, ist die Anwendung des Deduktionstheorems korrekt. $\checkmark$

8.2 Korrektheit von $\vdash_{\Box\mathsf{Fre}}$ für Modallogik

Lemma 8.5 (Korrektheitslemma für $\vdash_{\Box\mathsf{Fre}}$)

Sei α eine modallogische Formel.
Aus $\vdash_{\Box\mathsf{Fre}} \alpha$ folgt $\models_{K} \alpha$.

Beweis:

Wir erweitern den Beweis des Korrektheitslemmas für $\vdash_{\mathsf{Fre}}$ (5.2)
von „Belegung $\mathcal{A}$ erfüllt α“ auf „Kripke-Modell $\mathcal{M}$ erfüllt α in Welt u“.

Die Argumentation im Beweis von Lemma (5.2) geht damit genauso.

Zusätzlich muss folgendes gezeigt werden.

1. Im Induktionsanfang muss gezeigt werden, dass (K) gültig ist.
2. Im Induktionsschluss muss die Korrektheit der Regel Gen gezeigt werden.

zu 1.) Erweiterung des Induktionsanfangs:

Wir zeigen die Gültigkeit von (K).

Sei $\mathcal{M} = (W, R, \xi)$ ein Kripke-Modell und $u \in W$.

$$
\begin{aligned}
&\mathcal{M}, u \models_K \Box(\alpha \to \beta) \to (\Box\alpha \to \Box\beta) \\
&\quad\Leftrightarrow\quad \mathcal{M}, u \not\models_K \Box(\alpha \to \beta) \text{ oder } \mathcal{M}, u \not\models_K \Box\alpha \text{ oder } \mathcal{M}, u \models_K \Box\beta \\
&\quad\Leftrightarrow\quad \exists v, (u, v) \in R : \mathcal{M}, v \not\models_K \alpha \to \beta \text{ oder } \exists v, (u, v) \in R : \mathcal{M}, v \not\models_K \alpha \\
&\qquad\qquad \text{oder } \mathcal{M}, u \models_K \Box\beta \\
&\quad\Leftrightarrow\quad \exists v, (u, v) \in R : ((\mathcal{M}, v \models_K \alpha \text{ und } \mathcal{M}, v \not\models_K \beta) \text{ oder } \mathcal{M}, v \not\models_K \alpha) \\
&\qquad\qquad \text{oder } \mathcal{M}, u \models_K \Box\beta \\
&\quad\Leftrightarrow\quad \exists v, (u, v) \in R : (\mathcal{M}, v \not\models_K \beta \text{ oder } \mathcal{M}, v \not\models_K \alpha) \\
&\qquad\qquad \text{oder } \mathcal{M}, u \models_K \Box\beta \\
&\quad\Leftrightarrow\quad \exists v, (u, v) \in R : \mathcal{M}, v \not\models_K \alpha \text{ oder } \mathcal{M}, u \not\models_K \Box\beta \text{ oder } \mathcal{M}, u \models_K \Box\beta \\
&\quad\Leftrightarrow\quad \mathcal{M}, u \models_K \Box\beta \text{ oder } \mathcal{M}, u \not\models_K \Box\beta \quad \text{(wahre Aussage)} \qquad (\checkmark)
\end{aligned}
$$

zu 2.) Erweiterung des Induktionsschritts für Gen:

α sei mit einer Herleitung der Länge $k+1$ herleitbar,
und α entsteht durch Anwendung der Regel Gen auf ein β,
das mit $\leqslant k$ Schritten herleitbar ist.
Also ist $\alpha = \Box\beta$, und zu zeigen ist: $\models_K \Box\beta$.

Wähle $\mathcal{M} = (U, R, \xi)$ und $v \in U$ beliebig.

Gemäß IV ($\models_K \beta$) gilt $\mathcal{M}, u \models_K \beta$ für alle u mit $(v, u) \in R$.

Gemäß der Semantik von $\Box$ folgt $\mathcal{M}, v \models_K \Box\beta$.

Da $\mathcal{M}$ und v beliebig gewählt wurden, folgt $\models_K \Box\beta$. ✓

8.3 Vollständigkeit von $\vdash_{\Box\mathsf{Fre}}$ für Modallogik

Zum Beweis der Vollständigkeit von $\vdash_{\Box\mathsf{Fre}}$ benutzen wir die Vollständigkeit von $\vdash_{\Box\mathsf{Tab}}$ und die Umwandlung von modalen Tableau-Beweisen in modale Frege-Beweise – das entspricht dem Vorgehen im nicht-modalen Fall (siehe Beweis von Lemma 5.7).

Lemma 8.6 (aus $\neg\Box\beta$ expandierte Formeln sind überflüssig)

Seien $\alpha_1, \ldots, \alpha_m, \beta$ modallogische Formeln.
Aus $\alpha_1, \alpha_2, \ldots, \alpha_m, \neg\beta \vdash_{\Box\mathsf{Fre}} \bot$ folgt $\Box\alpha_1, \Box\alpha_2, \ldots, \Box\alpha_m, \neg\Box\beta \vdash_{\Box\mathsf{Fre}} \bot$.

(1) $\alpha_1, \alpha_2, \ldots, \alpha_m, \neg\beta \vdash_{\Box\mathsf{Fre}} \bot$ Voraussetzung

(2) $\vdash_{\Box\mathsf{Fre}} \alpha_1 \to (\alpha_2 \to (\ldots \to (\alpha_m \to (\neg\beta \to \bot)) \ldots))$ DT (1)

(3) $\vdash_{\Box\mathsf{Fre}} \Box(\alpha_1 \to (\alpha_2 \to (\ldots \to (\alpha_m \to \neg\neg\beta) \ldots)))$ Gen (2)

(4) $\vdash_{\Box\mathsf{Fre}} \Box(\alpha_1 \to (\ldots \to (\alpha_m \to \neg\neg\beta) \ldots)) \to \Box\alpha_1 \to (\ldots \to (\Box\alpha_m \to \Box\neg\neg\beta) \ldots))$

Lemma (8.4)

(5) $\vdash_{\Box\mathsf{Fre}} \Box\alpha_1 \to (\Box\alpha_2 \to (\ldots \to (\Box\alpha_m \to \Box\neg\neg\beta) \ldots))$ MP (3) (4)

(6) $\vdash_{\Box\mathsf{Fre}} \Box\neg\neg\beta \to \neg\neg\Box\beta$ Lemma (8.2)

(7) $\vdash_{\Box\mathsf{Fre}} \Box\alpha_1 \to (\Box\alpha_2 \to (\ldots \to (\Box\alpha_m \to \neg\neg\Box\beta) \ldots))$

Lemma (4.12) mit (5) (6)

(8) $\Box\alpha_1, \Box\alpha_2, \ldots, \Box\alpha_m \vdash_{\Box\mathsf{Fre}} \neg\Box\beta \to \bot$ DT (7)

(9) $\Box\alpha_1, \ldots, \Box\alpha_m, \neg\Box\beta \vdash_{\Box\mathsf{Fre}} \bot$ DT (8)

✓

Um das Lemma (5.7) entsprechende Lemma für modale Tableaux formulieren zu können, brauchen wir noch ein paar Begriffe.

Sei $\pi = v_1, \alpha_1, \ldots, v_\ell, \alpha_\ell$ ein Pfad durch ein Tableau T.
Er bestimmt einen Graph (W_π, R_π).
W_π ist die Menge aller Welten,
die in Markierungen von Knoten von π vorkommen.
R_π ist die Menge aller Kanten zwischen Welten,
die in Markierungen von Kanten in T vorkommen.

Wir nennen eine Welt $v \in W_\pi$ widersprüchlich,
wenn π für eine Formel α die Knoten v, α und $v, \neg\alpha$
oder den Knoten $v, \perp$ enthält.

Ein Knoten v, α heißt Startknoten von v,
wenn er nicht durch Expansion eines Knotens $v, \ldots$ entstanden ist.

$S_v = \{\alpha \mid v, \alpha \text{ ist Startknoten von } v\}$ ist die Menge der Startformeln von v.

Beispiel:

$W_{\pi_1} = \{s, t\}$
$R_{\pi_1} = \{(s, t)\}$

t ist widersprüchlich für π_1.

$S_s = \{\neg(\Box(A \to B) \to (\Box A \to \Box B))\}$

$S_t = \{A, A \to B, \neg B\}$

$$\begin{tikzcd}
& s, \neg(\Box(A \to B) \to (\Box A \to \Box B)) \arrow[d] & \\
& s, \Box(A \to B) \arrow[d] & \\
& s, \neg(\Box A \to \Box B) \arrow[d] & \\
& s, \Box A \arrow[d] & \\
& s, \neg\Box B \arrow[d, "{(s, t)}"] & \\
& t, \neg B \arrow[d] & \\
& t, A \to B \arrow[d] & \\
& t, A \arrow[dl] \arrow[dr] & \\
t, \neg A & & t, B \\
\times & & \times
\end{tikzcd}$$

Lemma 8.7 (Frege-Herleitungen von $\bot$ aus widersprüchlichen Tableaux)

Sei φ eine modallogische Formel und T ein widersprüchliches Tableau für φ.

Für jeden Pfad $\pi = w_1, \alpha_1, \ldots, \underbrace{w_j, \alpha_j}_{\pi_j}, \ldots, w_\ell, \alpha_\ell$ durch T gilt:

für jedes $i = 1, 2, \ldots, \ell$ gilt:
wenn man von w_i in (W_π, R_π) eine widersprüchliche Welt erreichen kann,
dann gilt $\quad S_{w_i} \cup \{\alpha_j \mid j \leqslant i \text{ und } \underbrace{w_i, \alpha_j}_{=\pi_j} \text{ kommt in } \pi \text{ vor}\} \vdash_{\Box\mathsf{Fre}} \bot$.

Beweis:
Sei T ein widersprüchliches Tableau.

Wir führen den Beweis für alle Pfade durch T mittels Induktion über $i = m, m-1, \ldots, 1$, wobei m die Länge des längsten Pfades durch T ist.

IA: i ist die Länge m eines längsten Pfades $\pi = \pi_1, \ldots, \pi_m$ durch T.
Sei $\pi_m = v, \alpha$.
Da T widersprüchlich ist, ist $\alpha = \bot$ oder
es gibt einen Knoten $\pi_j = v, \beta$ mit $j < m$, so dass $\beta = \neg\alpha$ oder $\alpha = \neg\beta$.
Dann gilt $\alpha, \beta \vdash_{\overline{\Box\mathsf{Fre}}} \bot$ und damit

$$S_v \cup \{\alpha_j \mid j \leqslant m \text{ und } v, \alpha_j \text{ kommt in } \pi \text{ vor}\} \vdash_{\overline{\Box\mathsf{Fre}}} \bot.$$

IV: Für k und jeden Pfad $\pi = w_1, \alpha_1, \ldots, w_\ell, \alpha_\ell$ durch T gilt:
für jedes $i = k, k+1, \ldots, \ell$ gilt:
wenn man von w_i in (W_π, R_π) eine widersprüchliche Welt erreichen kann,
gilt $S_{w_i} \cup \{\alpha_j \mid j \leqslant i \text{ und } w_i, \alpha_j \text{ kommt in } \pi \text{ vor}\} \vdash_{\overline{\Box\mathsf{Fre}}} \bot$.

IS: zu zeigen ist:
Für jeden Pfad $\pi = w_1, \alpha_1, \ldots, w_\ell, \alpha_\ell$ durch T gilt:
wenn man von w_{k-1} in (W_π, R_π) eine widersprüchliche Welt erreichen kann,
gilt $S_{w_{k-1}} \cup \{\alpha_j \mid j \leqslant k-1 \text{ und } w_{k-1}, \alpha_j \text{ kommt in } \pi \text{ vor}\} \vdash_{\overline{\Box\mathrm{Fre}}} \bot$.

Sei $\pi = w_1, \alpha_1, \ldots, w_\ell, \alpha_\ell$ ein Pfad durch T, und w_{k-1} sei eine Welt, von der aus man in (W_π, R_π) eine widersprüchliche Welt erreichen kann.

Fall (1): $w_k = w_{k-1}$
(d.h. der nächste Knoten auf π bezieht sich auf die gleiche Welt),
dann ist $\alpha_k \in S_{w_{k-1}}$
oder w_k, α_k entsteht durch Expansion von w_{k-1}, α_j für ein $j \leqslant k-1$
gemäß einer der beiden „aussagenlogischen" Expansionsregeln.
In diesem Fall argumentieren wir wie im Beweis von Lemma (5.7)
damit, dass die aus $\neg(\beta \to \gamma)$ oder $\beta \to \gamma$ expandierten Formeln
zur Herleitung von $\bot$ überflüssig sind
und erhalten aus der IV

$S_{w_{k-1}} \cup \{\alpha_j \mid j \leqslant k \text{ und } w_{k-1}, \alpha_j \text{ kommt in } \pi \text{ vor}\} \vdash_{\overline{\Box\mathrm{Fre}}} \bot$, dass
$S_{w_{k-1}} \cup \{\alpha_j \mid j \leqslant k-1 \text{ und } w_{k-1}, \alpha_j \text{ kommt in } \pi \text{ vor}\} \vdash_{\overline{\Box\mathrm{Fre}}} \bot$.

Fall (2): $w_k \neq w_{k-1}$.
Dann ist w_{k-1}, α_{k-1} der letzte Knoten mit w_{k-1} auf π,
und w_{k-1} hat in (W_π, R_π) einen Nachfolger u,
über den eine widersprüchliche Welt erreicht werden kann.
Also gibt es $w_{k-1}, \Box\beta_1, \ldots, w_{k-1}, \Box\beta_q, w_{k-1}, \neg\Box\gamma$ weiter vorne auf π,
mit $S_u = \{\beta_1, \ldots, \beta_q, \neg\gamma\}$.

Da Knoten mit u erst hinter Knoten mit w_{k-1} auf π vorkommen
und von u aus eine widersprüchliche Welt erreicht werden kann,
folgt $S_u \vdash_{\overline{\Box\mathrm{Fre}}} \bot$ aus der IV – also $\beta_1, \ldots, \beta_q, \neg\gamma \vdash_{\overline{\Box\mathrm{Fre}}} \bot$.

Mit Lemma (8.6) folgt dann $\Box\beta_1, \ldots, \Box\beta_q, \neg\Box\gamma \vdash_{\overline{\Box\mathrm{Fre}}} \bot$.

Da $\{\Box\beta_1, \ldots, \Box\beta_q, \neg\Box\gamma\} \subseteq$
$$S_{w_{k-1}} \cup \{\alpha_j \mid j \leqslant k-1 \text{ und } w_{k-1}, \alpha_j \text{ kommt in } \pi \text{ vor}\},$$
folgt $S_{w_{k-1}} \cup \{\alpha_j \mid j \leqslant k-1 \text{ und } w_{k-1}, \alpha_j \text{ kommt in } \pi \text{ vor}\} \vdash_{\overline{\Box\mathrm{Fre}}} \bot$.

$\checkmark$

Satz 8.8 (aus Tableau-Beweisen können Frege-Beweise gemacht werden)

Sei α eine modallogische Formel. Aus $\vdash_{\Box\mathrm{Tab}} \alpha$ folgt $\vdash_{\Box\mathrm{Fre}} \alpha$.

Beweis:

Sei $\vdash_{\Box\mathrm{Tab}} \alpha$.
Dann gibt es ein widersprüchliches Tableau T für $\neg\alpha$.
Sei $w, \neg\alpha$ der Anfangsknoten von T.
Dann ist $S_{w_1} = \{\neg\alpha\}$,
und der Fall $i = 1$ von Lemma (8.7) liefert $\neg\alpha \vdash_{\Box\mathrm{Fre}} \bot$.
Mit dem Deduktionstheorem (8.3) folgt dann $\vdash_{\Box\mathrm{Fre}} \neg\neg\alpha$, und daraus erhalten wir $\vdash_{\Box\mathrm{Fre}} \alpha$. ✓

8.4 Die Vollständigkeitssätze

Wie in der Aussagenlogik können wir nun auch für die Modallogik
aus der Korrektheit des Frege-Kalküls (8.5),
der Vollständigkeit des Tableau-Kalküls (7.6) und
der Möglichkeit, Tableau-Beweise in Frege-Beweise umzuwandeln (8.7)
die Vollständigkeitssätze für die beiden Beweis-Kalküle beweisen.

Wir fassen diesmal alles in einem Satz zusammen.

Satz 8.9 (Vollständigkeitssätze für die modale Aussagenlogik)

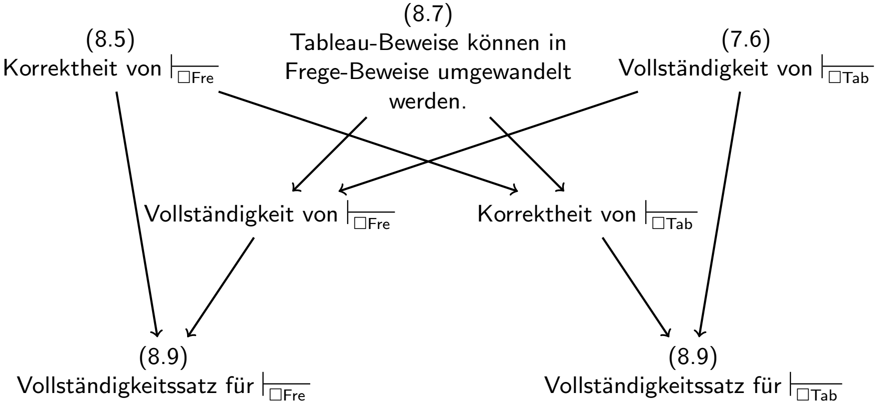
Sei φ eine modallogische Formel. Dann sind die folgenden Aussagen äquivalent.

1. $\models_{K} \varphi$
2. $\vdash_{\Box\mathrm{Tab}} \varphi$
3. $\vdash_{\Box\mathrm{Fre}} \varphi$

Was haben wir in Vorlesung 8 gelernt?

Wir haben den modallogischen Frege-Kalkül kennengelernt.

- Wir kennen die Axiome und Schlussregeln des modallogischen Frege-Kalküls.
- Wir können Frege-Beweise für modallogische Formeln führen und kennen $\vdash_{\Box\mathsf{Fre}}$.
- Wir wissen, dass das Deduktionstheorem in der Modallogik anders ist als in der Aussagenlogik.
- Wir wissen, dass Axiom (K) und Schlussregel Gen korrekt sind.
- Wir wissen, wie man Tableau-Beweise in Frege-Beweise umwandelt.
- Wir können den Vollständigkeitssatz für die betrachteten modallogischen Kalküle beweisen.



(8.5)
Korrektheit von $\vdash_{\Box\mathrm{Fre}}$
(8.7)
Tableau-Beweise können in Frege-Beweise umgewandelt werden.
(7.6)
Vollständigkeit von $\vdash_{\Box\mathrm{Tab}}$
Vollständigkeit von $\vdash_{\Box\mathrm{Fre}}$
Korrektheit von $\vdash_{\Box\mathrm{Tab}}$
(8.9)
Vollständigkeitssatz für $\vdash_{\Box\mathrm{Fre}}$
(8.9)
Vollständigkeitssatz für $\vdash_{\Box\mathrm{Tab}}$

Vorlesung 9: Algorithmische Umsetzung des Tableau-Kalküls und Ausblick

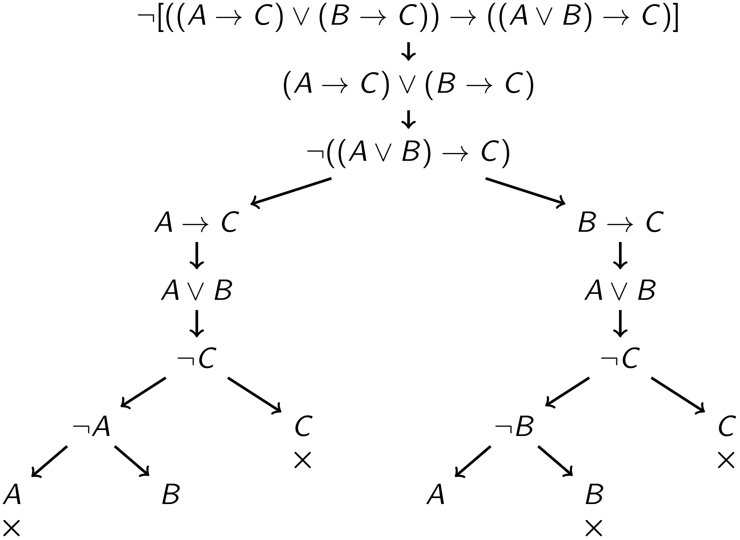
Vorlesung 9: Algorithmische Umsetzung des Tableau-Kalküls

Wir betrachten Algorithmen, die sich aus den Tableau-Kalkülen für Aussagenlogik und modale Aussagenlogik ergeben.

- Die Suche nach einem geschlossenen Pfad in einem aussagenlogischen Tableau lässt sich direkt als nichtdeterministischer Algorithmus ausdrücken.
- Nichtdeterminismus lässt sich deterministisch als rekursive Maximum-Suche formulieren.
- Die Suche nach einem geschlossenen Pfad in einem modallogischen Tableau lässt sich als rekursive Maximum/Minimum-Suche formulieren.
- Die Korrektheits- und Vollständigkeitsbeweise für die Tableau-Kalküle liefern uns, dass die Algorithmen
  - eine korrekte Ausgabe liefern und
  - bei jeder Eingabe eine Ausgabe liefern.

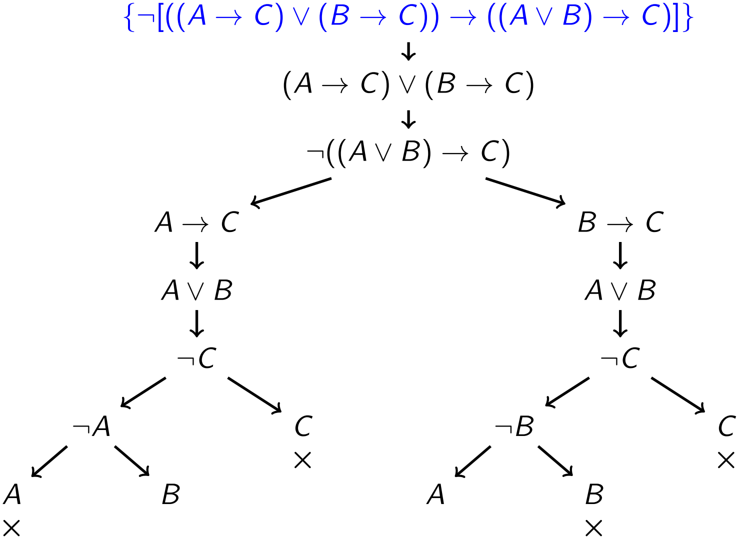
9.1 Algorithmen für die Aussagenlogik

Pfade durch Tableaux werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



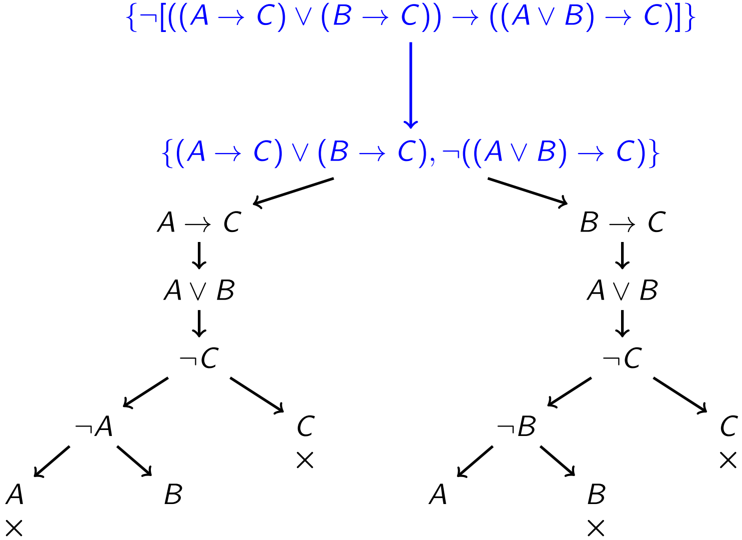
9.1 Algorithmen für die Aussagenlogik

Pfade durch Tableaux werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



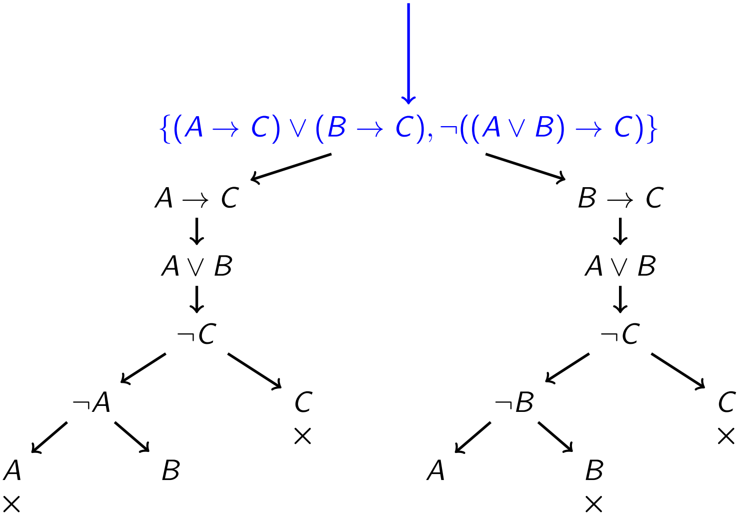
9.1 Algorithmen für die Aussagenlogik

Pfade durch Tableaux werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



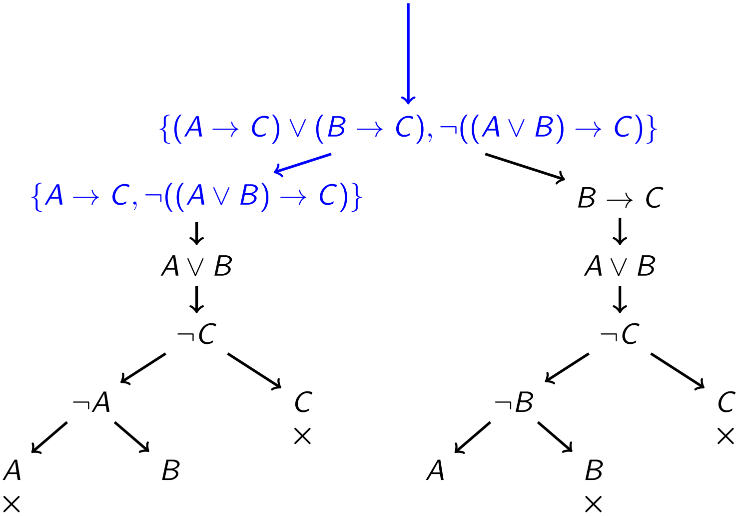
9.1 Algorithmen für die Aussagenlogik

Pfade durch Tableaux werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



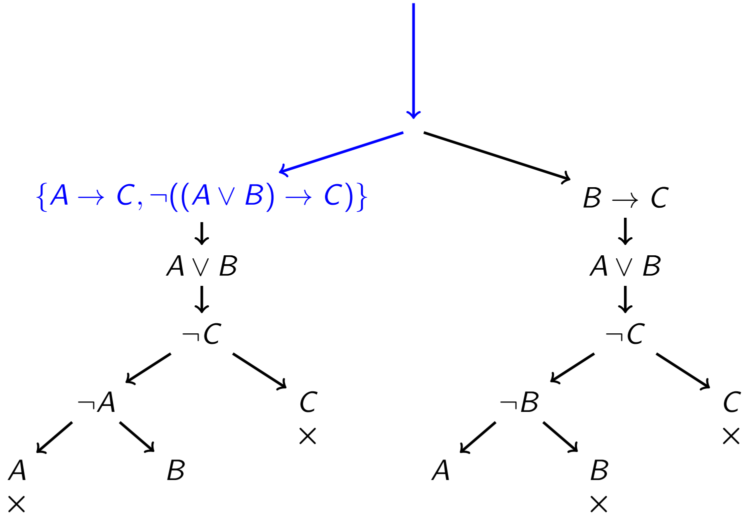
9.1 Algorithmen für die Aussagenlogik

Pfade durch Tableaux werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



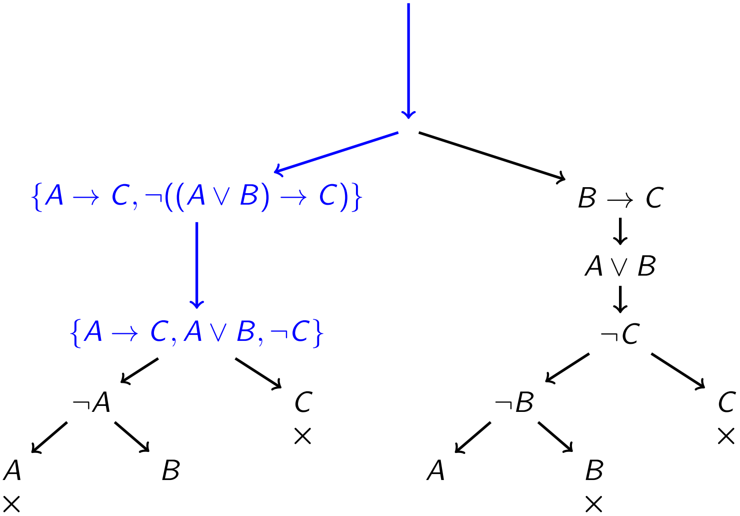
9.1 Algorithmen für die Aussagenlogik

Pfade durch Tableaux werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



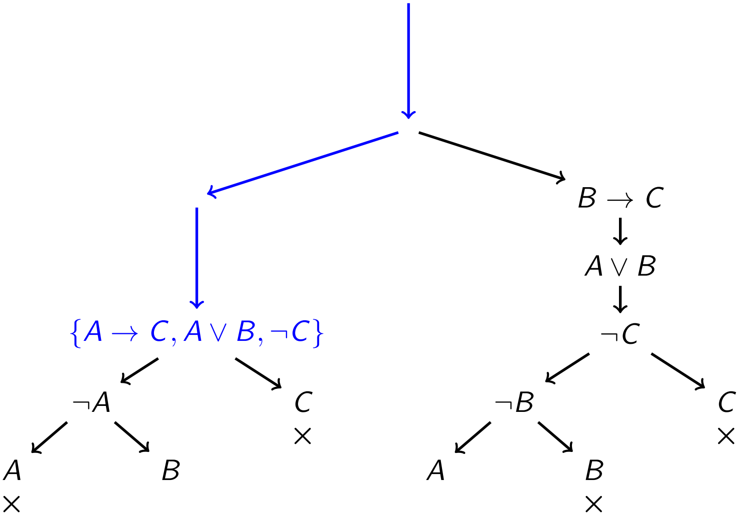
9.1 Algorithmen für die Aussagenlogik

Pfade durch Tableaux werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



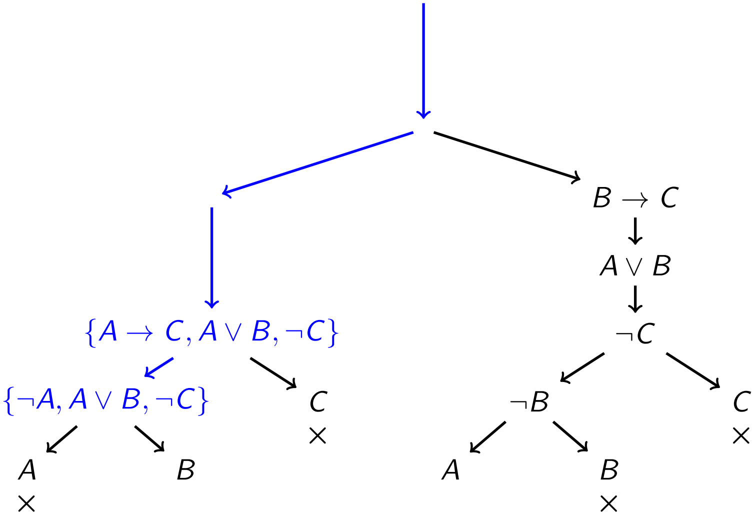
9.1 Algorithmen für die Aussagenlogik

Pfade durch Tableaux werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



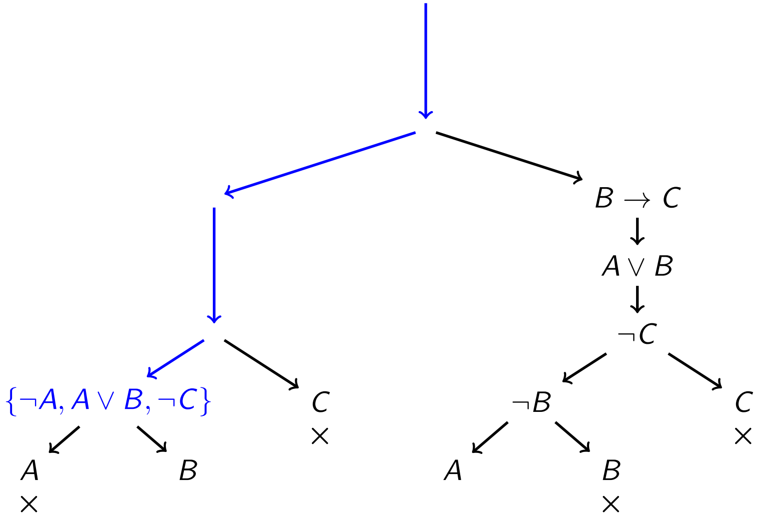
9.1 Algorithmen für die Aussagenlogik

Pfade durch Tableaux werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



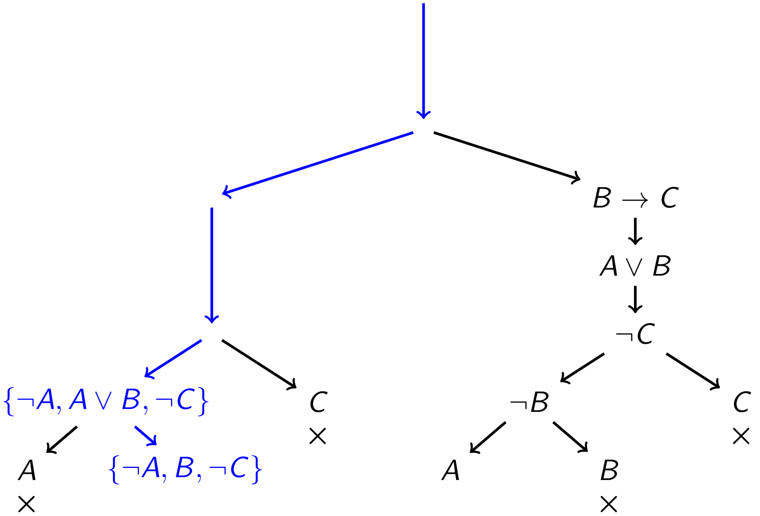
9.1 Algorithmen für die Aussagenlogik

Pfade durch Tableaux werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



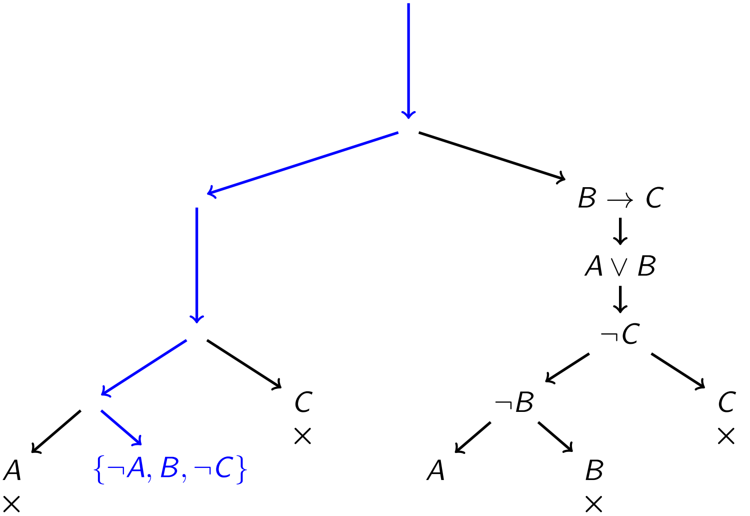
9.1 Algorithmen für die Aussagenlogik

Pfade durch Tableaux werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



9.1 Algorithmen für die Aussagenlogik

Pfade durch Tableaux werden durch Mengen von Formeln, die auf ihnen noch nicht expandiert wurden, dargestellt.



Jedes Tableau bestimmt einen Formelmengenbaum

$$
\begin{tikzcd}[column sep=tiny]
 & & & & \{\neg[((A \to C) \vee (B \to C)) \to ((A \vee B) \to C)]\} \arrow[d] & & & \\
 & & & & \{(A \to C) \vee (B \to C), \neg((A \vee B) \to C)\} \arrow[dll] \arrow[drr] & & & \\
 & & \{A \to C, \neg((A \vee B) \to C)\} \arrow[d] & & & & \{B \to C, \neg((A \vee B) \to C)\} \arrow[d] & \\
 & & \{A \to C, A \vee B, \neg C\} \arrow[dl] \arrow[d] & & & & \{B \to C, A \vee B, \neg C\} \arrow[dl] \arrow[d] & \\
 & \{\neg A, A \vee B, \neg C\} \arrow[dl] \arrow[d] & \{C, A \vee B, \neg C\} & & & \{\neg B, A \vee B, \neg C\} \arrow[dl] \arrow[d] & \{C, A \vee B, \neg C\} & \\
\{\neg A, A, \neg C\} & \{\neg A, B, \neg C\} & \times & & \{\neg B, A, \neg C\} & \{\neg B, B, \neg C\} & \times & \\
\times & & & & & \times & &
\end{tikzcd}
$$

Jede Menge repräsentiert einen Präfix eines Pfades durch das Tableau.

Jedes Tableau bestimmt einen Formelmengenbaum

$$
\begin{tikzcd}[column sep=tiny]
 & & & & \{\neg[((A \to C) \lor (B \to C)) \to ((A \lor B) \to C)]\} \arrow[d] & & & \\
 & & & & \{(A \to C) \lor (B \to C), \neg((A \lor B) \to C)\} \arrow[dll] \arrow[drr] & & & \\
 & & \{A \to C, \neg((A \lor B) \to C)\} \arrow[d] & & & & \{B \to C, \neg((A \lor B) \to C)\} \arrow[d] & \\
 & & \{A \to C, A \lor B, \neg C\} \arrow[dl] \arrow[dr] & & & & \{B \to C, A \lor B, \neg C\} \arrow[dl] \arrow[dr] & \\
 & \{\neg A, A \lor B, \neg C\} \arrow[dl] \arrow[dr] & & \begin{array}{c}\{C, A \lor B, \neg C\} \\ 0\end{array} & & \{\neg B, A \lor B, \neg C\} \arrow[dl] \arrow[dr] & & \begin{array}{c}\{C, A \lor B, \neg C\} \\ 0\end{array} \\
\begin{array}{c}\{\neg A, A, \neg C\} \\ 0\end{array} & & \{\neg A, B, \neg C\} & & \{\neg B, A, \neg C\} & & \begin{array}{c}\{\neg B, B, \neg C\} \\ 0\end{array} &
\end{tikzcd}
$$

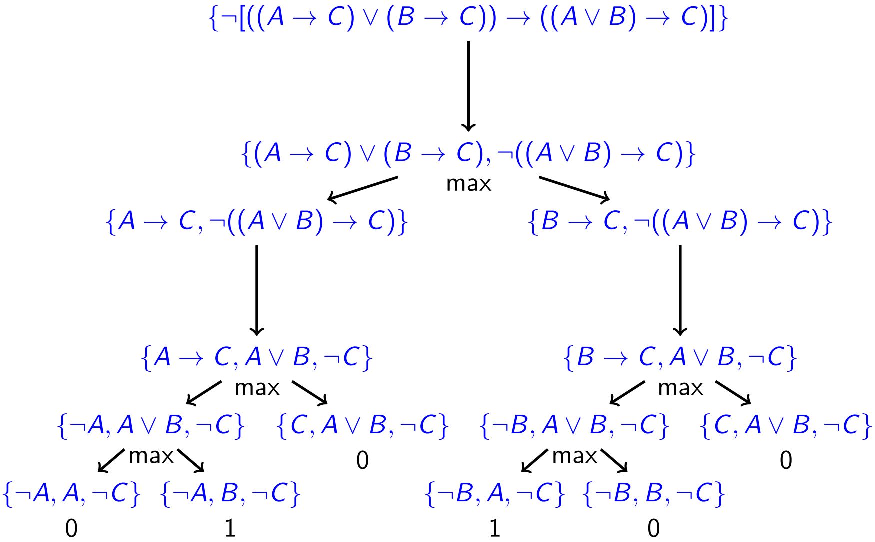
Jede Menge repräsentiert einen Präfix eines Pfades durch das Tableau.

Jedes Tableau bestimmt einen Formelmengenbaum

$$\begin{tikzcd}[column sep=tiny]
 & & & & \{\neg[((A \to C) \vee (B \to C)) \to ((A \vee B) \to C)]\} \arrow[d] & & & \\
 & & & & \{(A \to C) \vee (B \to C), \neg((A \vee B) \to C)\} \arrow[dll] \arrow[drr] & & & \\
 & & \{A \to C, \neg((A \vee B) \to C)\} \arrow[d] & & & & \{B \to C, \neg((A \vee B) \to C)\} \arrow[d] & \\
 & & \{A \to C, A \vee B, \neg C\} \arrow[dl] \arrow[dr] & & & & \{B \to C, A \vee B, \neg C\} \arrow[dl] \arrow[dr] & \\
 & \{\neg A, A \vee B, \neg C\} \arrow[dl] \arrow[dr] & & \begin{array}{c}\{C, A \vee B, \neg C\} \\ 0\end{array} & & \{\neg B, A \vee B, \neg C\} \arrow[dl] \arrow[dr] & & \begin{array}{c}\{C, A \vee B, \neg C\} \\ 0\end{array} \\
\begin{array}{c}\{\neg A, A, \neg C\} \\ 0\end{array} & & \begin{array}{c}\{\neg A, B, \neg C\} \\ 1\end{array} & & \begin{array}{c}\{\neg B, A, \neg C\} \\ 1\end{array} & & \begin{array}{c}\{\neg B, B, \neg C\} \\ 0\end{array} &
\end{tikzcd}$$

Jede Menge repräsentiert einen Präfix eines Pfades durch das Tableau.

Jedes Tableau bestimmt einen Formelmengenbaum



Jede Menge repräsentiert einen Präfix eines Pfades durch das Tableau.

Idee für einen Algorithmus

Stelle mittels Tiefensuche durch den Formelmengenbaum fest,
dass alle Pfade des Tableaus widersprüchlich sind.

Das geht genauso wie:

Stelle mittels Tiefensuche durch den Formelmengenbaum fest,
dass es einen geschlossenen (nicht-widersprüchlichen) Pfad
durch das Tableau gibt.

Gültigkeitstest gemäß Tableau-Kalkül

für aussagenlogische Formeln mittels einer rekursiven Methode für die Tiefensuche durch den Formelmengenbaum

Methode erfüllbar(Formelmenge $\mathcal{S}$):

(* liefert Ergebnis 1, falls $\mathcal{S}$ erfüllbar ist, und Ergebnis 0 sonst *)

falls $\mathcal{S}$ widersprüchlich ist: return 0 (* $\mathcal{S}$ ist widersprüchlich *)

falls $\mathcal{S}$ eine Formel ψ der Form $\neg(\alpha \to \beta)$ enthält:

(* ersetze ψ in $\mathcal{S}$ durch alle ihre Zerlegungsformeln *)

return erfüllbar$((\mathcal{S} - \psi) \cup \{\alpha, \neg\beta\})$

sonst: falls $\mathcal{S}$ eine Formel ψ der Form $\alpha \to \beta$ mit $\beta \neq \bot$ enthält:

(* ersetze ψ in $\mathcal{S}$ „parallel" durch jeweils eine Zerlegungsformel *)

return $\max_{\gamma \in \{\neg\alpha, \beta\}}$ erfüllbar$((\mathcal{S} - \psi) \cup \{\gamma\})$

sonst: (* $\mathcal{S}$ ist nicht widersprüchlich und enthält nur noch Literale *)

return 1 (* $\mathcal{S}$ ist erfüllende Belegung von φ *)

Tableau-Algorithmus (liefert Ergebnis 1, falls φ gültig ist, und Erg. 0 sonst)

Eingabe Formel φ

Ausgabe $1 -$ erfüllbar$(\{\neg\varphi\})$

Grobe Analyse des Gültigkeitstests

für Aussagenlogik

Der Algorithmus simuliert das Tableau-Verfahren.

Jeder Tiefensuchepfad entspricht einem Pfad durch das Tableau.

Die Menge $\mathcal{S}$ enthält stets die Knoten-Markierungen auf dem Pfad, die noch nicht expandiert wurden.

Da jede Formel der Länge n höchstens $2n$ Teilformeln besitzt, wird jeder Tiefensuchepfad in polynomieller Zeit durchlaufen,

Da das Tableau für eine Formel der Länge n höchsten 2^n Pfade hat, hat der Algorithmus exponentielle Rechenzeit.

Das gleiche als nichtdeterministischer Algorithmus

nichtdeterministischer Erfüllbarkeitstest gemäß Tableau-Verfahren:

Eingabe: Formelmenge $\mathcal{S}$

(∗ es gibt einen akzeptierenden Berechnungspfad gdw. $\mathcal{S}$ erfüllbar ist ∗)

solange $\mathcal{S}$ nicht widersprüchlich und expandierbar ist, wiederhole:

falls $\mathcal{S}$ eine Formel ψ der Form $\neg(\alpha \to \beta)$ enthält:

(∗ ersetze ψ durch alle ihre Zerlegungsformeln ∗)

$\mathcal{S} = (\mathcal{S} - \psi) \cup \{\alpha, \neg\beta\}$

sonst: falls $\mathcal{S}$ eine Formel ψ der Form $\alpha \to \beta$ mit $\beta \neq \bot$ enthält:

(∗ wähle nichtdeterministisch eine Zerlegungsformel ∗)

wähle (existentiell) nichtdeterministisch $\gamma \in \{\neg\alpha, \beta\}$

$\mathcal{S} = (\mathcal{S} - \psi) \cup \{\gamma\}$

falls $\mathcal{S}$ nicht widersprüchlich ist: akzeptiere (∗ entspricht Ausgabe 1 ∗)

sonst: verwirf (∗ entspricht Ausgabe 0 ∗)

Es gilt:

(1) der nichtdetermistische Algorithmus hat bei Eingabe $\{\varphi\}$ einen akzeptierenden Berechnungspfad genau dann, wenn φ erfüllbar ist.

(2) der nichtdeterministische Algorithmus hat polynomielle Rechenzeit.

9.2 Algorithmen für Modallogik

Bei der Modallogik kommt noch die Navigation zwischen den Welten hinzu.

Die Kripke-Modelle, die von Pfaden durch die Tableaux bestimmt werden, sind Bäume.

Wir konstruieren einen rekursiven Algorithmus, bei dem die Struktur der rekursiven Aufrufe der Baum-Struktur des Kripke-Modells entspricht.

Das führt dazu, dass der Algorithmus das Kripke-Modell ignorieren kann.

Was kommt bei Modallogik dazu?

$$
\begin{tikzcd}[column sep=small]
& s, \neg(\Box(A \to B) \to (\Box A \to \Box B)) \arrow[d] & \\
& s, \Box(A \to B) \arrow[d] & \\
& s, \neg(\Box A \to \Box B) \arrow[d] & \\
& s, \Box A \arrow[d] & \\
& s, \neg\Box B \arrow[d, "{(s, t)}"] & \\
& t, \neg B \arrow[d] & \\
& t, A \to B \arrow[d] & \\
& t, A \arrow[dl] \arrow[dr] & \\
t, \neg A & & t, B \\
\times & & \times
\end{tikzcd}
$$

$$
\begin{tikzcd}[column sep=small]
& s, \neg(\Box(A \to B) \to (\Box A \to \Box B)) \arrow[d] & \\
& s, \Box(A \to B) \arrow[d] & \\
& s, \neg(\Box A \to \Box B) \arrow[d] & \\
& s, \Box A \arrow[d] & \\
& s, \neg\Box B \arrow[d, "{(s, t) \in R}"] & \\
& t, \neg B \arrow[d] & \\
& t, A \to B \arrow[d] & \\
& t, A \arrow[dl] \arrow[dr] & \\
t, \neg A & & t, B \\
\times & & \times
\end{tikzcd}
$$

Was kommt bei Modallogik dazu?

$$
\begin{tikzcd}[column sep=small]
 & s, \neg(\Box(A \to B) \to (\Box A \to \Box B)) \arrow[d] & \\
 & s, \Box(A \to B) \arrow[d] & \\
 & s, \neg(\Box A \to \Box B) \arrow[d] & \\
 & s, \Box A \arrow[d] & \\
 & s, \neg\Box B \arrow[d, "{(s,t)}"] & \\
 & t, \neg B \arrow[d] & \\
 & t, A \to B \arrow[d] & \\
 & t, A \arrow[dl] \arrow[dr] & \\
t, \neg A & & t, B \\
\times & & \times
\end{tikzcd}
$$

$$
\begin{tikzcd}[column sep=small]
 & \{\neg(\Box(A \to B) \to (\Box A \to \Box B))\} \arrow[d] & \\
 & s, \Box(A \to B) \arrow[d] & \\
 & s, \neg(\Box A \to \Box B) \arrow[d] & \\
 & s, \Box A \arrow[d] & \\
 & s, \neg\Box B \arrow[d, "{(s,t) \in R}"] & \\
 & t, \neg B \arrow[d] & \\
 & t, A \to B \arrow[d] & \\
 & t, A \arrow[dl] \arrow[dr] & \\
t, \neg A & & t, B \\
\times & & \times
\end{tikzcd}
$$

Was kommt bei Modallogik dazu?

$$
\begin{tikzcd}
 & s, \neg(\Box(A \to B) \to (\Box A \to \Box B)) \arrow[d] & \\
 & s, \Box(A \to B) \arrow[d] & \\
 & s, \neg(\Box A \to \Box B) \arrow[d] & \\
 & s, \Box A \arrow[d] & \\
 & s, \neg\Box B \arrow[d, "{(s, t)}"] & \\
 & t, \neg B \arrow[d] & \\
 & t, A \to B \arrow[d] & \\
 & t, A \arrow[dl] \arrow[dr] & \\
t, \neg A & & t, B \\
\times & & \times
\end{tikzcd}
$$

$$
\begin{tikzcd}
 & \{\neg(\Box(A \to B) \to (\Box A \to \Box B))\} \arrow[d] & \\
 & \{\Box(A \to B), \neg(\Box A \to \Box B)\} \arrow[d] & \\
 & s, \Box A \arrow[d] & \\
 & s, \neg\Box B \arrow[d, "{(s, t) \in R}"] & \\
 & t, \neg B \arrow[d] & \\
 & t, A \to B \arrow[d] & \\
 & t, A \arrow[dl] \arrow[dr] & \\
t, \neg A & & t, B \\
\times & & \times
\end{tikzcd}
$$

Was kommt bei Modallogik dazu?

$$
\begin{tikzcd}
 & s, \neg(\Box(A \to B) \to (\Box A \to \Box B)) \arrow[d] & \\
 & s, \Box(A \to B) \arrow[d] & \\
 & s, \neg(\Box A \to \Box B) \arrow[d] & \\
 & s, \Box A \arrow[d] & \\
 & s, \neg\Box B \arrow[d, "(s,t)"] & \\
 & t, \neg B \arrow[d] & \\
 & t, A \to B \arrow[d] & \\
 & t, A \arrow[dl] \arrow[dr] & \\
t, \neg A & & t, B \\
\times & & \times
\end{tikzcd}
$$

$$
\begin{tikzcd}
 & \{\neg(\Box(A \to B) \to (\Box A \to \Box B))\} \arrow[d] & \\
 & \{\Box(A \to B), \neg(\Box A \to \Box B)\} \arrow[d] & \\
 & \{\Box(A \to B), \Box A, \neg\Box B\} \arrow[d, "(s,t) \in R"] & \\
 & t, \neg B \arrow[d] & \\
 & t, A \to B \arrow[d] & \\
 & t, A \arrow[dl] \arrow[dr] & \\
t, \neg A & & t, B \\
\times & & \times
\end{tikzcd}
$$

Was kommt bei Modallogik dazu?

$$
\begin{tikzcd}[column sep=small]
 & s, \neg(\Box(A \to B) \to (\Box A \to \Box B)) \arrow[d] & \\
 & s, \Box(A \to B) \arrow[d] & \\
 & s, \neg(\Box A \to \Box B) \arrow[d] & \\
 & s, \Box A \arrow[d] & \\
 & s, \neg\Box B \arrow[d, "{(s,t)}"] & \\
 & t, \neg B \arrow[d] & \\
 & t, A \to B \arrow[d] & \\
 & t, A \arrow[dl] \arrow[dr] & \\
t, \neg A & & t, B \\
\times & & \times
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
 & \{\neg(\Box(A \to B) \to (\Box A \to \Box B))\} \arrow[d] & \\
 & \{\Box(A \to B), \neg(\Box A \to \Box B)\} \arrow[d] & \\
 & \{\Box(A \to B), \Box A, \neg\Box B\} \arrow[d, "{(s,t) \in R}"] & \\
 & \{A \to B, A, \neg B\} \arrow[dl] \arrow[dr] & \\
t, \neg A & & t, B \\
\times & & \times
\end{tikzcd}
$$

Was kommt bei Modallogik dazu?

$$
\begin{tikzcd}
 & s, \neg(\Box(A \to B) \to (\Box A \to \Box B)) \arrow[d] & \\
 & s, \Box(A \to B) \arrow[d] & \\
 & s, \neg(\Box A \to \Box B) \arrow[d] & \\
 & s, \Box A \arrow[d] & \\
 & s, \neg\Box B \arrow[d, "{(s,t)}"] & \\
 & t, \neg B \arrow[d] & \\
 & t, A \to B \arrow[d] & \\
 & t, A \arrow[dl] \arrow[dr] & \\
t, \neg A & & t, B \\
\times & & \times
\end{tikzcd}
$$

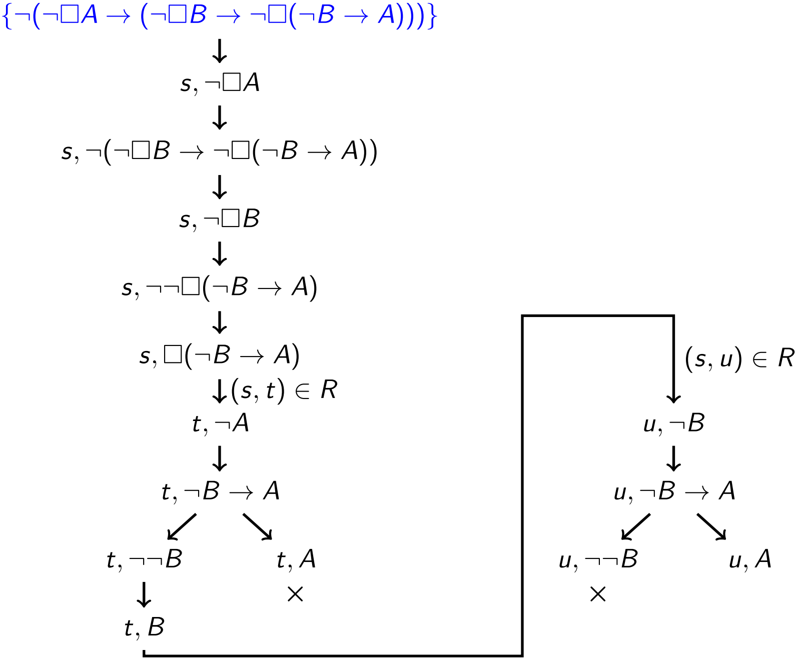
$$
\begin{tikzcd}
 & \{\neg(\Box(A \to B) \to (\Box A \to \Box B))\} \arrow[d] & \\
 & \{\Box(A \to B), \neg(\Box A \to \Box B)\} \arrow[d] & \\
 & \{\Box(A \to B), \Box A, \neg\Box B\} \arrow[d, "{(s,t) \in R}"] & \\
 & \{A \to B, A, \neg B\} \arrow[dl] \arrow[dr] & \\
\{\neg A, A, \neg B\} & & \{B, A, \neg B\} \\
\times & & \times
\end{tikzcd}
$$

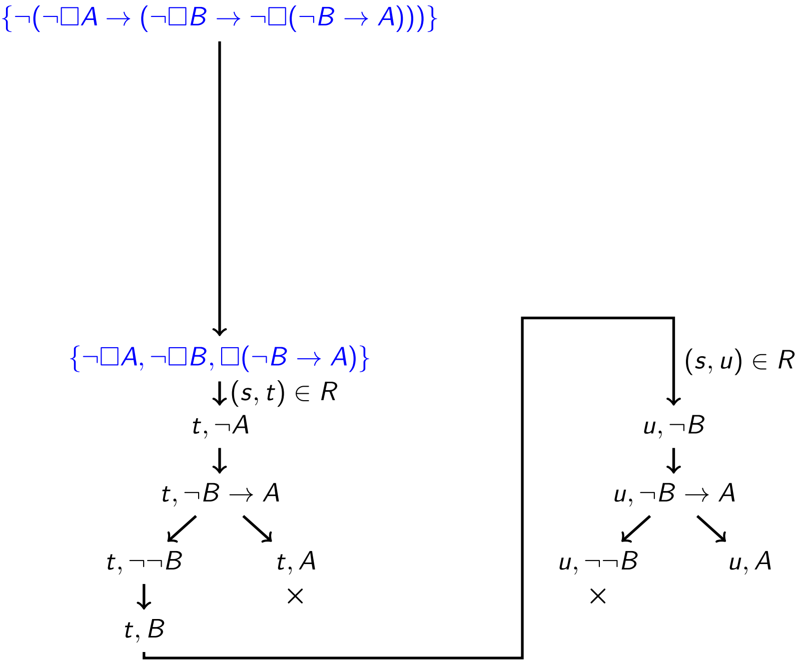
Was kommt bei Modallogik dazu?

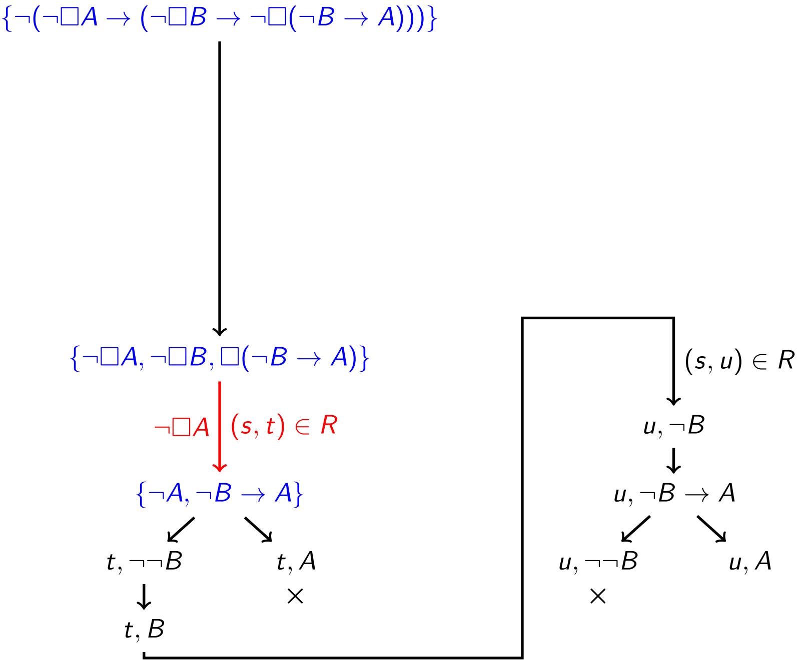
$$
\begin{array}{c}
s, \neg(\Box(A \to B) \to (\Box A \to \Box B)) \\
\downarrow \\
s, \Box(A \to B) \\
\downarrow \\
s, \neg(\Box A \to \Box B) \\
\downarrow \\
s, \Box A \\
\downarrow \\
s, \neg\Box B \\
\Big\downarrow (s,t) \\
t, \neg B \\
\downarrow \\
t, A \to B \\
\downarrow \\
t, A \\
\swarrow \qquad \searrow \\
\begin{array}{cc}
t, \neg A & t, B \\
\times & \times
\end{array}
\end{array}
$$

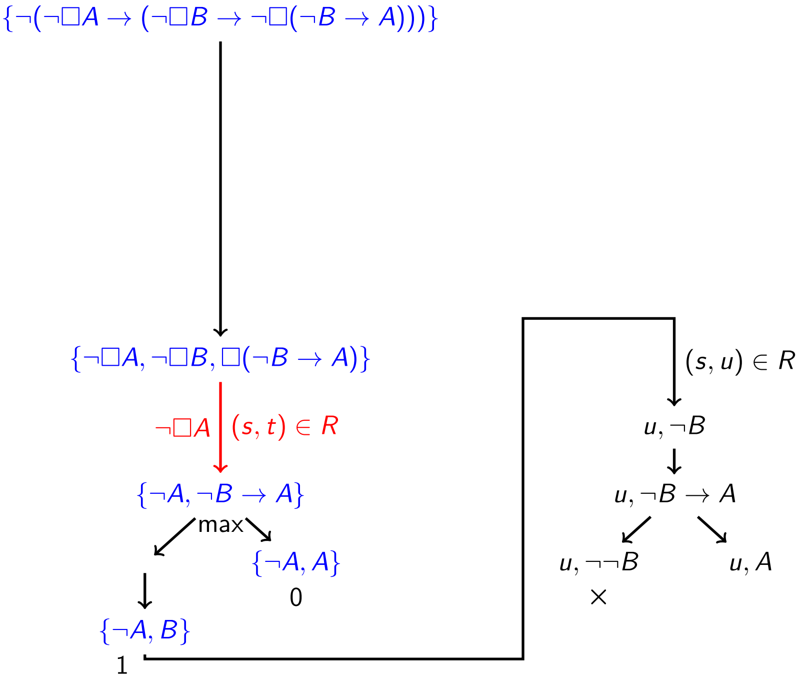
$$
\begin{array}{c}
\{\neg(\Box(A \to B) \to (\Box A \to \Box B))\} \\
\Big\downarrow \\
\{\Box(A \to B), \neg(\Box A \to \Box B)\} \\
\Big\downarrow \\
\{\Box(A \to B), \Box A, \neg\Box B\} \\
\Big\downarrow (s,t) \in R \\
\{A \to B, A, \neg B\} \\
\swarrow \ \text{max} \ \searrow \\
\begin{array}{cc}
\{\neg A, A, \neg B\} & \{B, A, \neg B\} \\
0 & 0
\end{array}
\end{array}
$$

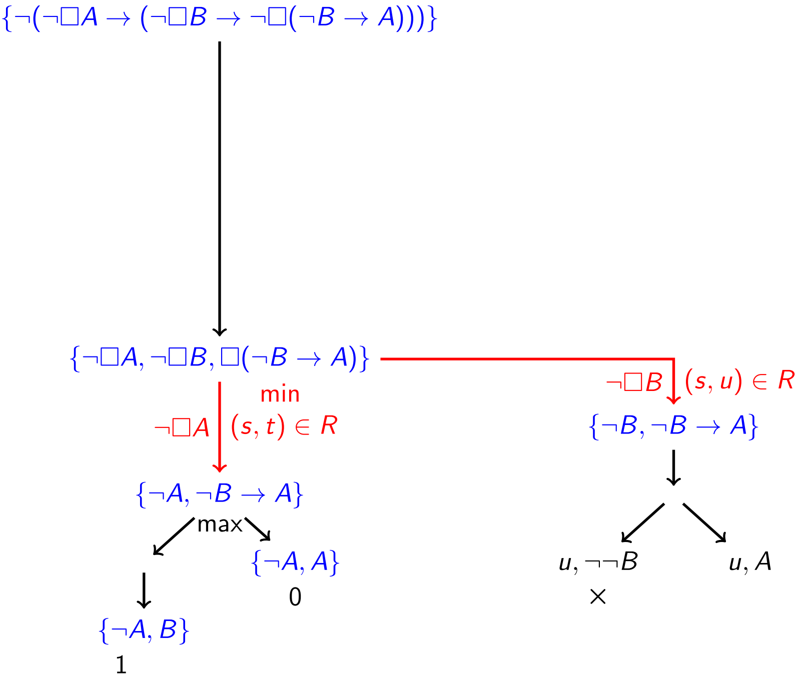
$$
\begin{tikzcd}[column sep=small]
 & s, \neg(\neg\Box A \to (\neg\Box B \to \neg\Box(\neg B \to A))) \arrow[d] & & & & \\
 & s, \neg\Box A \arrow[d] & & & & \\
 & s, \neg(\neg\Box B \to \neg\Box(\neg B \to A)) \arrow[d] & & & & \\
 & s, \neg\Box B \arrow[d] & & & & \\
 & s, \neg\neg\Box(\neg B \to A) \arrow[d] & & & & \\
 & s, \Box(\neg B \to A) \arrow[d, "{(s,t) \in R}"] & & & & \\
 & t, \neg A \arrow[d] & & & u, \neg B \arrow[d] & \\
 & t, \neg B \to A \arrow[dl] \arrow[dr] & & & u, \neg B \to A \arrow[dl] \arrow[dr] & \\
t, \neg\neg B \arrow[dd] & & t, A & u, \neg\neg B & & u, A \\
 & & \times & \times & & \\
t, B \arrow[uuuurrrr, bend right, "{(s,u) \in R}"'] & & & & &
\end{tikzcd}
$$

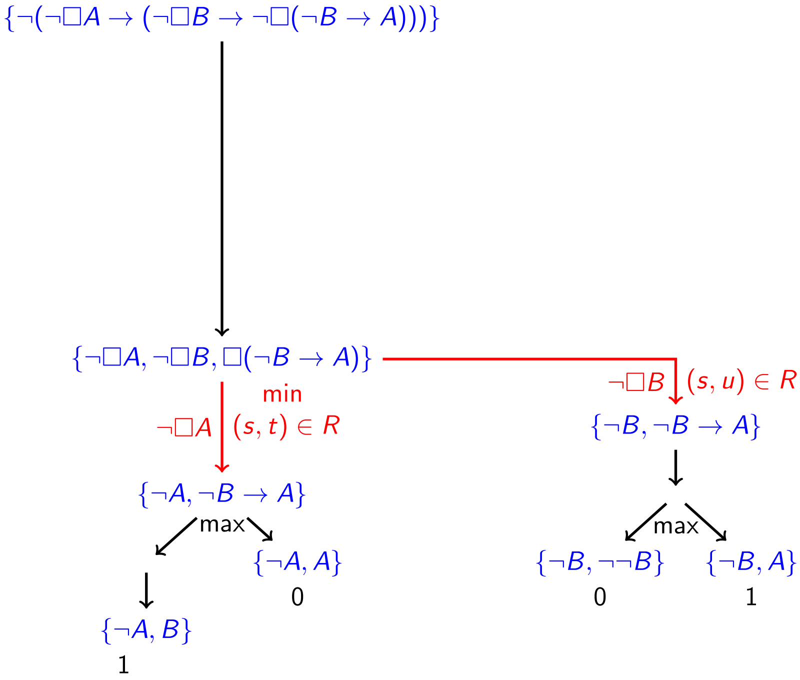

$\{\neg(\neg\Box A \to (\neg\Box B \to \neg\Box(\neg B \to A)))\}$
$s, \neg\Box A$
$s, \neg(\neg\Box B \to \neg\Box(\neg B \to A))$
$s, \neg\Box B$
$s, \neg\neg\Box(\neg B \to A)$
$s, \Box(\neg B \to A)$
$(s, t) \in R$
$t, \neg A$
$t, \neg B \to A$
$t, \neg\neg B$
t, A
$\times$
t, B
$(s, u) \in R$
$u, \neg B$
$u, \neg B \to A$
$u, \neg\neg B$
u, A
$\times$


{¬(¬□A → (¬□B → ¬□(¬B → A)))}
{¬□A, ¬□B, □(¬B → A)}
(s, t) ∈ R
t, ¬A
t, ¬B → A
t, ¬¬B
t, A
×
t, B
(s, u) ∈ R
u, ¬B
u, ¬B → A
u, ¬¬B
u, A
×


$\{\neg(\neg\Box A \to (\neg\Box B \to \neg\Box(\neg B \to A)))\}$
$\{\neg\Box A, \neg\Box B, \Box(\neg B \to A)\}$
$\neg\Box A$
$(s,t) \in R$
$\{\neg A, \neg B \to A\}$
$t, \neg\neg B$
t, A
$\times$
t, B
$(s,u) \in R$
$u, \neg B$
$u, \neg B \to A$
$u, \neg\neg B$
u, A
$\times$


$\{\neg(\neg\Box A \to (\neg\Box B \to \neg\Box(\neg B \to A)))\}$
$\{\neg\Box A, \neg\Box B, \Box(\neg B \to A)\}$
$\neg\Box A$
$(s,t) \in R$
$\{\neg A, \neg B \to A\}$
max
$\{\neg A, A\}$
0
$\{\neg A, B\}$
1
$(s,u) \in R$
$u, \neg B$
$u, \neg B \to A$
$u, \neg\neg B$
$\times$
u, A


{¬(¬□A → (¬□B → ¬□(¬B → A)))}
{¬□A, ¬□B, □(¬B → A)}
min
¬□A
(s, t) ∈ R
¬□B
(s, u) ∈ R
{¬A, ¬B → A}
max
{¬A, A}
0
{¬A, B}
1
{¬B, ¬B → A}
u, ¬¬B
×
u, A


$\{\neg(\neg\Box A \to (\neg\Box B \to \neg\Box(\neg B \to A)))\}$
$\{\neg\Box A, \neg\Box B, \Box(\neg B \to A)\}$
min
$\neg\Box A$
$(s,t) \in R$
$\neg\Box B$
$(s,u) \in R$
$\{\neg A, \neg B \to A\}$
max
$\{\neg A, A\}$
0
$\{\neg A, B\}$
1
$\{\neg B, \neg B \to A\}$
max
$\{\neg B, \neg\neg B\}$
0
$\{\neg B, A\}$
1

Gültigkeitstest gemäß Tableau-Kalkül

für modallogische Formeln als rekursive Methode

Methode merfüllbar(Formelmenge $\mathcal{S}$):

(∗ liefert Ergebnis 1, falls $\mathcal{S}$ unerfüllbar ist, und Ergebnis 0 sonst ∗)

falls $\mathcal{S}$ widersprüchlich ist: return 0

falls $\mathcal{S}$ eine Formel ψ der Form $\neg(\alpha \to \beta)$ enthält:

return merfüllbar$((\mathcal{S} - \psi) \cup \{\alpha, \neg\beta\})$

sonst: falls $\mathcal{S}$ eine Formel ψ der Form $\alpha \to \beta$ mit $\beta \neq \bot$ enthält:

return $\max\limits_{\gamma \in \{\neg\alpha, \beta\}}$ merfüllbar$((\mathcal{S} - \psi) \cup \{\gamma\})$

sonst: falls $\mathcal{S}$ eine Formel ψ der Form $\neg\Box\varphi$ enthält:

return $\min\limits_{\neg\Box\beta \in \mathcal{S}}$ merfüllbar$(\{\alpha \mid \Box\alpha \in \mathcal{S}\} \cup \{\neg\beta\})$

sonst: (∗ $\mathcal{S}$ ist nicht widersprüchlich und muss nicht weiter expandiert werden ∗)

return 1

Modaler Tableau-Algorithmus (∗ liefert Ergebnis 1, falls φ gültig ist, und Erg. 0 sonst ∗)

Eingabe Formel φ

Ausgabe 1 − merfüllbar$(\{\neg\varphi\})$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend me$[X, Y, Z]$ für merfüllbar$(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

$$\text{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)]$$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend $\mathrm{me}[X, Y, Z]$ für merfüllbar$(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

$$
\begin{tikzcd}[column sep=tiny]
& \mathrm{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)] \arrow[dl, "\neg\Box\neg A"'] \arrow[d, phantom, "\text{min}"] \arrow[dr, "\neg\Box A"] & \\
\mathrm{me}[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] & {} & \mathrm{me}[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]
\end{tikzcd}
$$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend me$[X, Y, Z]$ für merfüllbar$(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

$$
\begin{tikzcd}[column sep=tiny]
 & & \mathrm{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)] \arrow[dl, "\neg\Box\neg A"'] \arrow[dr, "\neg\Box A"] \arrow[d, phantom, "\mathrm{min}"] & \\
 & \mathrm{me}[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[dl, "\neg A"'] \arrow[d, "\Box A"] \arrow[d, phantom, "\mathrm{max}\ \ A \to \Box A"'] & \, & \mathrm{me}[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \\
\mathrm{me}[A, \neg A, \ldots] = 0 & \mathrm{me}[A, \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] & &
\end{tikzcd}
$$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend $\mathrm{me}[X, Y, Z]$ für $\mathrm{merfüllbar}(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

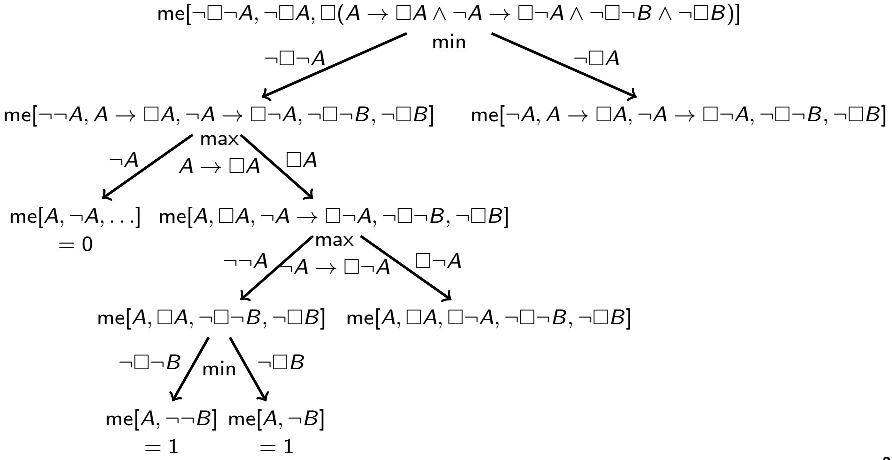
$$
\begin{tikzcd}[column sep=tiny]
 & & \underset{\text{min}}{\mathrm{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)]} \arrow[dl, "\neg\Box\neg A"'] \arrow[dr, "\neg\Box A"] & \\
 & \underset{\text{max}\quad A \to \Box A}{\mathrm{me}[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]} \arrow[dl, "\neg A"'] \arrow[dr, "\Box A"] & & \mathrm{me}[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \\
 \begin{matrix}\mathrm{me}[A, \neg A, \ldots] \\ = 0\end{matrix} & & \underset{\text{max}\quad \neg A \to \Box\neg A}{\mathrm{me}[A, \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]} \arrow[dl, "\neg\neg A"'] \arrow[dr, "\Box\neg A"] & \\
 & \mathrm{me}[A, \Box A, \neg\Box\neg B, \neg\Box B] & & \mathrm{me}[A, \Box A, \Box\neg A, \neg\Box\neg B, \neg\Box B]
\end{tikzcd}
$$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \land \neg\Box A \land \Box(A \to \Box A \land \neg A \to \Box\neg A \land \neg\Box\neg B \land \neg\Box B)\})$

Wir schreiben abkürzend $\mathrm{me}[X, Y, Z]$ für $\mathrm{merfüllbar}(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

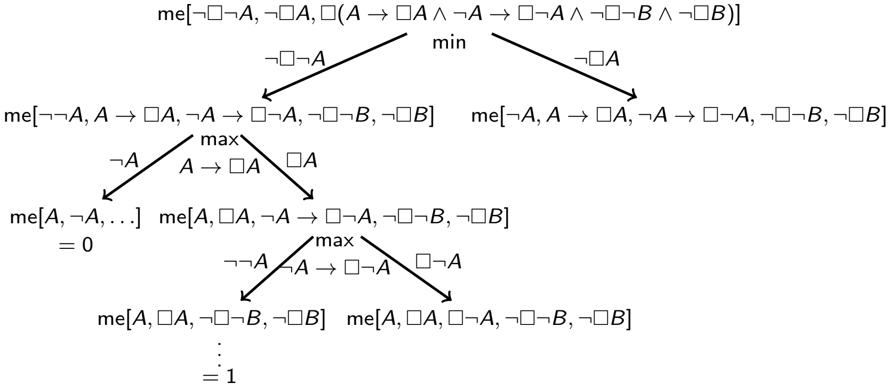


Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend me$[X, Y, Z]$ für merfüllbar$(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

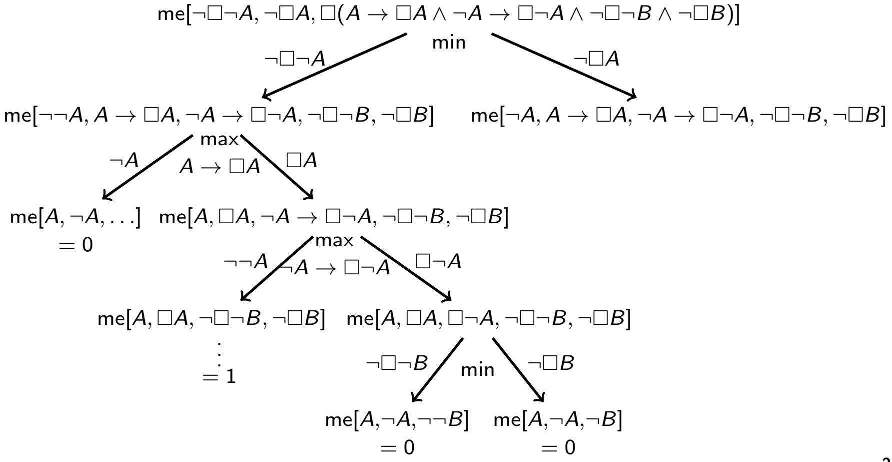


Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend me$[X, Y, Z]$ für merfüllbar$(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.



Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend me$[X, Y, Z]$ für merfüllbar$(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

$$\mathrm{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)]$$

min

$\xrightarrow{\neg\Box\neg A}$ $\mathrm{me}[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]$

$\xrightarrow{\neg\Box A}$ $\mathrm{me}[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]$

max, $A \to \Box A$

$\xrightarrow{\neg A}$ $\mathrm{me}[A, \neg A, \ldots] = 0$

$\xrightarrow{\Box A}$ $\mathrm{me}[A, \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]$

max, $\neg A \to \Box\neg A$

$\xrightarrow{\neg\neg A}$ $\mathrm{me}[A, \Box A, \neg\Box\neg B, \neg\Box B] \;\vdots\; = 1$

$\xrightarrow{\Box\neg A}$ $\mathrm{me}[A, \Box A, \Box\neg A, \neg\Box\neg B, \neg\Box B] \;\vdots\; = 0$

Struktur der rekursiven Aufrufe

für merfüllbar($\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\}$)

Wir schreiben abkürzend me$[X, Y, Z]$ für merfüllbar($\{X, Y, Z\}$).

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

$$
\begin{tikzcd}
& \mathrm{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)] \arrow[dl, "\neg\Box\neg A"'] \arrow[dr, "\neg\Box A"] \arrow[d, phantom, "\mathrm{min}"] & \\
\mathrm{me}[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, dotted] & & \mathrm{me}[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \\
= 1 & &
\end{tikzcd}
$$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend $\mathrm{me}[X, Y, Z]$ für $\mathrm{merfüllbar}(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

$$
\begin{tikzcd}[column sep=small]
 & \mathrm{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)] \arrow[dl, "\neg\Box\neg A"'] \arrow[dr, "\neg\Box A"] \arrow[d, phantom, "\mathrm{min}"] & & \\
\mathrm{me}[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, dotted] & & \mathrm{me}[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[dl, "\neg A"'] \arrow[dr, "\Box A"] \arrow[d, phantom, "\mathrm{max}"] \arrow[d, phantom, "A \to \Box A"', shift left=6] & \\
= 1 & \mathrm{me}[\neg A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] & & \mathrm{me}[\neg A, \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]
\end{tikzcd}
$$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend me$[X, Y, Z]$ für merfüllbar$(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

$$
\begin{tikzcd}[column sep=tiny]
 & & \begin{array}{c}\text{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)] \\ \text{min}\end{array} \arrow[dl, "\neg\Box\neg A"'] \arrow[dr, "\neg\Box A"] & \\
 & \begin{array}{c}\text{me}[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \\ \vdots \\ = 1\end{array} & & \begin{array}{c}\text{me}[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \\ \text{max} \\ A \to \Box A\end{array} \arrow[dl, "\neg A"'] \arrow[d, "\Box A"] \\
 & & \begin{array}{c}\text{me}[\neg A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \\ \text{max} \\ \neg A \to \Box\neg A\end{array} \arrow[dll, "\neg\neg A"'] \arrow[d, "\Box\neg A"] & \text{me}[\neg A, \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \\
\begin{array}{c}\text{me}[\neg A, A, \neg\Box\neg B, \neg\Box B] \\ = 0\end{array} & & \text{me}[\neg A, \Box\neg A, \neg\Box\neg B, \neg\Box B] &
\end{tikzcd}
$$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend me$[X, Y, Z]$ für merfüllbar$(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

$$
\begin{tikzcd}[column sep=tiny]
& & \underset{\text{min}}{\mathrm{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)]} \arrow[dll, "\neg\Box\neg A"'] \arrow[dr, "\neg\Box A"] & & \\
\mathrm{me}[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, dotted] & & & \underset{\text{max} \quad A \to \Box A}{\mathrm{me}[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]} \arrow[dl, "\neg A"'] \arrow[dr, "\Box A"] & \\
= 1 & & \underset{\text{max} \quad \neg A \to \Box\neg A}{\mathrm{me}[\neg A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]} \arrow[dl, "\neg\neg A"'] \arrow[dr, "\Box\neg A"] & & \mathrm{me}[\neg A, \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \\
& \underset{= 0}{\mathrm{me}[\neg A, A, \neg\Box\neg B, \neg\Box B]} & & \underset{\text{min}}{\mathrm{me}[\neg A, \Box\neg A, \neg\Box\neg B, \neg\Box B]} \arrow[dl, "\neg\Box\neg B"'] \arrow[dr, "\neg\Box B"] & \\
& & \underset{= 1}{\mathrm{me}[\neg A, \neg\neg B]} & & \underset{= 1}{\mathrm{me}[\neg A, \neg B]}
\end{tikzcd}
$$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend me$[X, Y, Z]$ für merfüllbar$(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

- me$[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)]$ — min
  - $\xrightarrow{\neg\Box\neg A}$ me$[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]$ $\vdots$ $= 1$
  - $\xrightarrow{\neg\Box A}$ me$[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]$ — max, $A \to \Box A$
    - $\xrightarrow{\neg A}$ me$[\neg A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]$ — max, $\neg A \to \Box\neg A$
      - $\xrightarrow{\neg\neg A}$ me$[\neg A, A, \neg\Box\neg B, \neg\Box B]$ $= 0$
      - $\xrightarrow{\Box\neg A}$ me$[\neg A, \Box\neg A, \neg\Box\neg B, \neg\Box B]$ $\vdots$ $= 1$
    - $\xrightarrow{\Box A}$ me$[\neg A, \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend $\mathrm{me}[X, Y, Z]$ für $\mathrm{merfüllbar}(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

$$
\begin{tikzcd}
 & \mathrm{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)] \arrow[dl, "\neg\Box\neg A"'] \arrow[dr, "\neg\Box A"] \arrow[d, phantom, "\text{min}"] & & \\
\mathrm{me}[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, dotted, no head] & \; & \mathrm{me}[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[dl, "\neg A"'] \arrow[dr, "\Box A"] \arrow[d, phantom, "\text{max} \quad A \to \Box A"] & \\
= 1 & \mathrm{me}[\neg A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, dotted, no head] & \; & \mathrm{me}[\neg A, \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \\
 & = 1 & &
\end{tikzcd}
$$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend me$[X, Y, Z]$ für merfüllbar$(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

$$
\begin{tikzcd}[column sep=tiny]
& \mathrm{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)] \arrow[dl, "\neg\Box\neg A"'] \arrow[dr, "\neg\Box A"] \arrow[d, phantom, "\mathrm{min}"] & & \\
\mathrm{me}[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, dotted] & {} & \mathrm{me}[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[dl, "\neg A"'] \arrow[dr, "\Box A"] \arrow[d, phantom, "\mathrm{max}\; A \to \Box A"] & \\
= 1 & \mathrm{me}[\neg A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, dotted] & {} & \mathrm{me}[\neg A, \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[dl, "\neg\neg A"'] \arrow[d, "\Box\neg A"] \arrow[dl, phantom, "\mathrm{max}\; \neg A \to \Box\neg A"] \\
& = 1 & \mathrm{me}[\neg A, \Box A, A, \neg\Box\neg B, \neg\Box B] = 0 & \mathrm{me}[\neg A, \Box A, \Box\neg A, \neg\Box\neg B, \neg\Box B]
\end{tikzcd}
$$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend me$[X, Y, Z]$ für merfüllbar$(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

- me$[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)]$ — min
  - $\xrightarrow{\neg\Box\neg A}$ me$[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]$ $\vdots$ $= 1$
  - $\xrightarrow{\neg\Box A}$ me$[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]$ — max, $A \to \Box A$
    - $\xrightarrow{\neg A}$ me$[\neg A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]$ $\vdots$ $= 1$
    - $\xrightarrow{\Box A}$ me$[\neg A, \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B]$ — max, $\neg A \to \Box\neg A$
      - $\xrightarrow{\neg\neg A}$ me$[\neg A, \Box A, A, \neg\Box\neg B, \neg\Box B]$ $= 0$
      - $\xrightarrow{\Box\neg A}$ me$[\neg A, \Box A, \Box\neg A, \neg\Box\neg B, \neg\Box B]$ — min
        - $\xrightarrow{\neg\Box\neg B}$ me$[A, \neg A, \neg\neg B]$ $= 0$
        - $\xrightarrow{\neg\Box B}$ me$[A, \neg A, \neg B]$ $= 0$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend me$[X, Y, Z]$ für merfüllbar$(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

$$
\begin{tikzcd}[column sep=small]
 & \begin{array}{c}\text{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)] \\ \text{min}\end{array} \arrow[dl, "\neg\Box\neg A"'] \arrow[dr, "\neg\Box A"] & & \\
\text{me}[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, dotted] & & \begin{array}{c}\text{me}[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \\ \text{max}\end{array} \arrow[dl, "\neg A"'] \arrow[dr, "\Box A"] \arrow[d, phantom, "A \to \Box A"] & \\
= 1 & \text{me}[\neg A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, dotted] & & \begin{array}{c}\text{me}[\neg A, \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \\ \text{max}\end{array} \arrow[dl, "\neg\neg A"'] \arrow[d, "\Box\neg A"] \arrow[dl, phantom, "\neg A \to \Box\neg A", bend left] \\
 & = 1 & \begin{array}{c}\text{me}[\neg A, \Box A, A, \neg\Box\neg B, \neg\Box B] \\ = 0\end{array} & \text{me}[\neg A, \Box A, \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, dotted] \\
 & & & = 0
\end{tikzcd}
$$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend $\text{me}[X, Y, Z]$ für merfüllbar$(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

$$
\begin{tikzcd}[column sep=tiny]
 & \text{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)] \arrow[dl, "\neg\Box\neg A"'] \arrow[dr, "\neg\Box A"] \arrow[d, phantom, "\text{min}"] & & \\
\text{me}[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, dotted] & {} & \text{me}[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[dl, "\neg A"'] \arrow[dr, "\Box A"] \arrow[d, phantom, "\substack{\text{max} \\ A \to \Box A}"] & \\
= 1 & \text{me}[\neg A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, dotted] & {} & \text{me}[\neg A, \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, dotted] \\
 & = 1 & & = 0
\end{tikzcd}
$$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend me$[X, Y, Z]$ für merfüllbar$(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

$$
\begin{tikzcd}[column sep=small]
 & \mathrm{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)] \arrow[dl, "\neg\Box\neg A"'] \arrow[dr, "\neg\Box A"] \arrow[d, phantom, "\mathrm{min}"] & \\
\mathrm{me}[\neg\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, phantom, "\vdots"] & {} & \mathrm{me}[\neg A, A \to \Box A, \neg A \to \Box\neg A, \neg\Box\neg B, \neg\Box B] \arrow[d, phantom, "\vdots"] \\
= 1 & & = 1
\end{tikzcd}
$$

Struktur der rekursiven Aufrufe

für merfüllbar$(\{\neg\Box\neg A \wedge \neg\Box A \wedge \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)\})$

Wir schreiben abkürzend $\mathrm{me}[X, Y, Z]$ für $\mathrm{merfüllbar}(\{X, Y, Z\})$.

Rekursive Aufrufe für Formeln vom Typ 1 sind nicht aufgeführt.

$$\begin{array}{c} \mathrm{me}[\neg\Box\neg A, \neg\Box A, \Box(A \to \Box A \wedge \neg A \to \Box\neg A \wedge \neg\Box\neg B \wedge \neg\Box B)] \\ \vdots \\ = 1 \end{array}$$

(Grobe) Analyse

Es gilt: φ ist gültig genau dann, wenn merfüllbar$(\{\neg\varphi\}) = 0$.

Bei Aufruf von merfüllbar$(\{\alpha\})$ wird ein Baum durchsucht,

- dessen Tiefe durch die Anzahl der Verknüpfungszeichen in α beschränkt ist
- bei dem der Ausgangsgrad jedes Knotens durch die Anzahl der modalen Operatoren in α beschränkt ist
- in dem jeder Knoten mit einer Formelmenge markiert ist, deren Größe durch die Größe von α beschränkt ist.

Der Speicherverbrauch ist also $O(|\alpha|^2)$ für den Rekursionsstapel.

Satz 9.1

Das Erfüllbarkeitsproblem für modale Aussagenlogik ist in PSPACE.

Exkurs: Komplexität von Logikproblemen

Die Tabelle zeigt komplexitätstheoretische Vollständigkeitsresultate.

	Erfüllbarkeit	Gültigkeit	Formelauswertung
Aussagenlogik (AL)	NP [Cook 71]	coNP [Cook 71]	NC^1 [Buss 87]
Modale AL	PSPACE [Ladner 77]	PSPACE [Ladner 77]	P [Folklore]
LTL	PSPACE [Sistla,Clarke 85]	PSPACE [Sistla,Clarke 85]	PSPACE [Sistla,Clarke 85]

$$\begin{tikzcd}
 & & NP \arrow[dr] & \\
NC^1 \arrow[r] & P \arrow[ur] \arrow[dr] & & PSPACE \\
 & & coNP \arrow[ur] &
\end{tikzcd}$$

polynomielle Zeit — polynomieller Platz

parallel — deterministisch — nicht-determ.

superschnell — schnell — langsam

Was haben wir in Vorlesung 9 gelernt?

- Die Suche nach einem geschlossenen Pfad in einem aussagenlogischen Tableau ist ein NP-Algorithmus.
- Das Durchsuchen eines modallogischen Tableaus ist ein PSPACE-Algorithmus.

Exkurs und Abschluss von Kapitel 2

Andere Modallogiken

Wir betrachten ein Beispiel zur Modellierung mit Modallogik:
muddy children.

0. Drei Kinder haben draußen gespielt, und einige haben nun ein schmutziges Gesicht.
Jedes Kind sieht die anderen Kinder, kann aber nicht sein eigenes Gesicht sehen.

1. Der Vater sagt: „Mindestens einer von euch hat ein schmutziges Gesicht".

2. Der Vater fragt (1.Frage): „Wer weiß, ob er ein schmutziges Gesicht hat?"
Kein Kind meldet sich.

3. Der Vater fragt (2.Frage): „Wer weiß, ob er ein schmutziges Gesicht hat?"
Kind 1 und Kind 3 melden sich.

4. Der Vater fragt (3.Frage): „Wer weiß, ob er ein schmutziges Gesicht hat?"
Kind 2 meldet sich auch.

Kind 1 und Kind 3 sagen: „Ich habe ein schmutziges Gesicht!"
Kind 2 sagt: „Ich habe kein schmutziges Gesicht!"

Wie ist das zu erklären?

Das Wissen der Kinder vor der ersten Interaktion

$A\,B\,C$

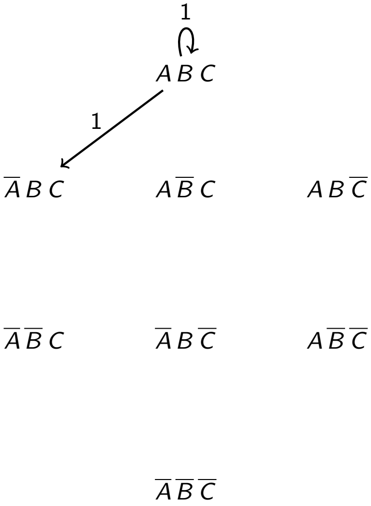
$\overline{A}\,B\,C$ $\quad$ $A\,\overline{B}\,C$ $\quad$ $A\,B\,\overline{C}$

$\overline{A}\,\overline{B}\,C$ $\quad$ $\overline{A}\,B\,\overline{C}$ $\quad$ $A\,\overline{B}\,\overline{C}$

$\overline{A}\,\overline{B}\,\overline{C}$

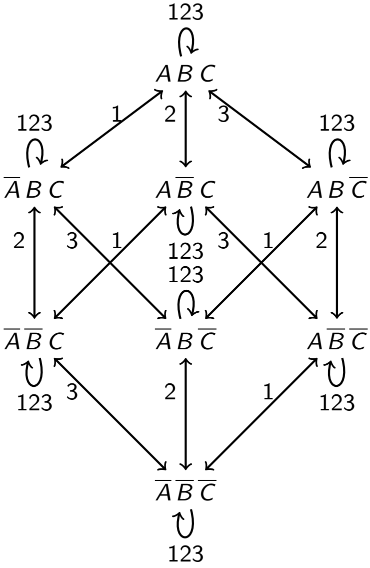
- Die möglichen Zustände/Welten:
  $A \triangleq$ Kind 1 hat schmutziges Gesicht
  $\overline{B} \triangleq$ Kind 2 hat kein schmutziges Gesicht
- Das Wissen der Kinder (am Anfang) über ihr eigenes schmutziges Gesicht: in Zustand ABC weiß Kind 1, dass Zustände ABC und $\overline{A}BC$ möglich sind.
- Die Belegung:
  Zustand $A\,\overline{B}\,C$ hat Belegung $\{A, C\}$
- Das ist ein Kripke-Modell für multimodale Logik:
  $\overline{A}\,B\,\overline{C} \models_K \Diamond_1 A \wedge \Box_2 \neg A$
- Bedeutung von $\Box_i$ und $\Diamond_i$:
  $w \models_K \Box_i A$ gdw. Kind i *weiß* in Zustand w, dass Kind 1 ein schmutziges Gesicht hat
  $w \models_K \Diamond_i C$ gdw. Kind i *hält es in Zustand w für möglich*, dass Kind 3 ein schmutziges Gesicht hat.

Das Wissen der Kinder vor der ersten Interaktion



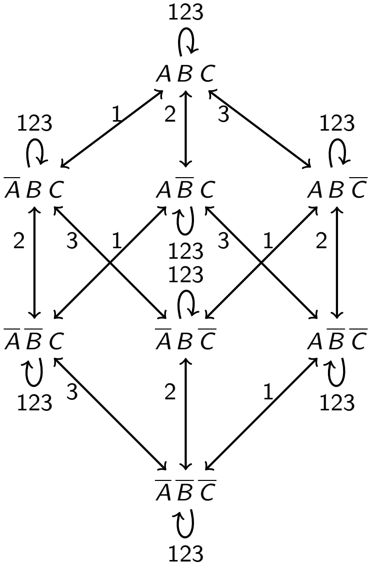
- Die möglichen Zustände/Welten:
  $A \mathrel{\hat{=}}$ Kind 1 hat schmutziges Gesicht
  $\overline{B} \mathrel{\hat{=}}$ Kind 2 hat kein schmutziges Gesicht
- Das Wissen der Kinder (am Anfang) über ihr eigenes schmutziges Gesicht: in Zustand ABC weiß Kind 1, dass Zustände ABC und $\overline{A}BC$ möglich sind.
- Die Belegung:
  Zustand $A\,\overline{B}\,C$ hat Belegung $\{A, C\}$
- Das ist ein Kripke-Modell für multimodale Logik:
  $\overline{A}\,B\,\overline{C} \models_K \Diamond_1 A \wedge \Box_2 \neg A$
- Bedeutung von $\Box_i$ und $\Diamond_i$:
  $w \models_K \Box_i A$ gdw. Kind i *weiß* in Zustand w, dass Kind 1 ein schmutziges Gesicht hat
  $w \models_K \Diamond_i C$ gdw. Kind i *hält es in Zustand w für möglich*, dass Kind 3 ein schmutziges Gesicht hat.

Das Wissen der Kinder vor der ersten Interaktion



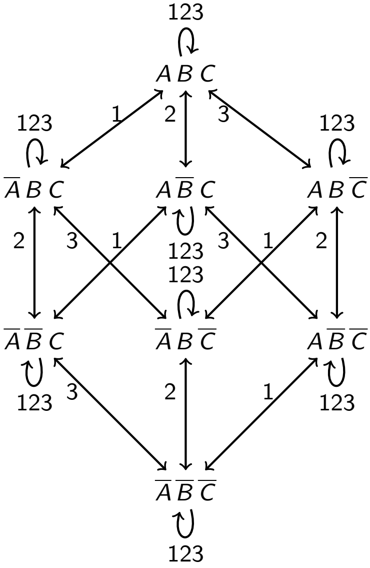
- Die möglichen Zustände/Welten:
  $A \mathrel{\hat{=}}$ Kind 1 hat schmutziges Gesicht
  $\overline{B} \mathrel{\hat{=}}$ Kind 2 hat kein schmutziges Gesicht
- Das Wissen der Kinder (am Anfang) über ihr eigenes schmutziges Gesicht: in Zustand ABC weiß Kind 1, dass Zustände ABC und $\overline{A}BC$ möglich sind.
- Die Belegung:
  Zustand $A\,\overline{B}\,C$ hat Belegung $\{A, C\}$
- Das ist ein Kripke-Modell für multimodale Logik:
  $\overline{A}\,B\,\overline{C} \models_K \Diamond_1 A \wedge \Box_2 \neg A$
- Bedeutung von $\Box_i$ und $\Diamond_i$:
  $w \models_K \Box_i A$ gdw. Kind i *weiß* in Zustand w, dass Kind 1 ein schmutziges Gesicht hat
  $w \models_K \Diamond_i C$ gdw. Kind i *hält es in Zustand w für möglich*, dass Kind 3 ein schmutziges Gesicht hat.

Das Wissen der Kinder vor der ersten Interaktion



- Die möglichen Zustände/Welten:
  $A \mathrel{\widehat{=}}$ Kind 1 hat schmutziges Gesicht
  $\overline{B} \mathrel{\widehat{=}}$ Kind 2 hat kein schmutziges Gesicht
- Das Wissen der Kinder (am Anfang) über ihr eigenes schmutziges Gesicht: in Zustand ABC weiß Kind 1, dass Zustände ABC und $\overline{A}BC$ möglich sind.
- Die Belegung:
  Zustand $A\,\overline{B}\,C$ hat Belegung $\{A, C\}$
- Das ist ein Kripke-Modell für multimodale Logik:
  $\overline{A}\,B\,\overline{C} \models_K \Diamond_1 A \land \Box_2 \neg A$
- Bedeutung von $\Box_i$ und $\Diamond_i$:
  $w \models_K \Box_i A$ gdw. Kind i *weiß* in Zustand w, dass Kind 1 ein schmutziges Gesicht hat
  $w \models_K \Diamond_i C$ gdw. Kind i *hält es in Zustand w für möglich*, dass Kind 3 ein schmutziges Gesicht hat.

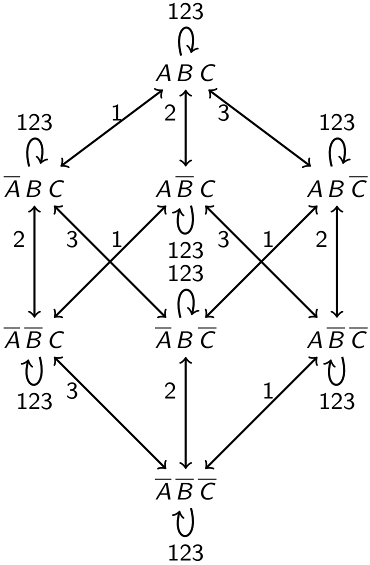
Das Wissen der Kinder vor der ersten Interaktion



- Die möglichen Zustände/Welten:
  $A \triangleq$ Kind 1 hat schmutziges Gesicht
  $\overline{B} \triangleq$ Kind 2 hat kein schmutziges Gesicht
- Das Wissen der Kinder (am Anfang) über ihr eigenes schmutziges Gesicht: in Zustand ABC weiß Kind 1, dass Zustände ABC und $\overline{A}BC$ möglich sind.
- Die Belegung:
  Zustand $A\,\overline{B}\,C$ hat Belegung $\{A, C\}$
- Das ist ein Kripke-Modell für multimodale Logik:
  $\overline{A}\,B\,\overline{C} \models_K \Diamond_1 A \wedge \Box_2 \neg A$
- Bedeutung von $\Box_i$ und $\Diamond_i$:
  $w \models_K \Box_i A$ gdw. Kind *i weiß* in Zustand w, dass Kind 1 ein schmutziges Gesicht hat
  $w \models_K \Diamond_i C$ gdw. Kind *i hält es in Zustand w für möglich*, dass Kind 3 ein schmutziges Gesicht hat.

Das Wissen der Kinder vor der ersten Frage

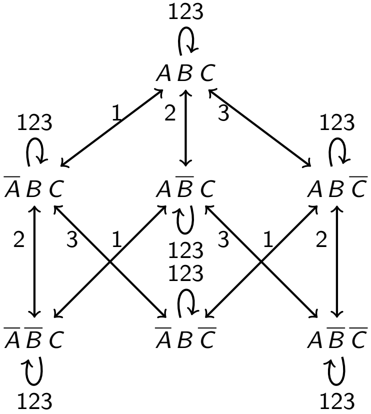
„Mindestens ein Kind hat ein schmutziges Gesicht“



- Zustand $\overline{A}\,\overline{B}\,\overline{C}$ ist nicht möglich.
- $A\,\overline{B}\,\overline{C} \models_K \Box_1 A \vee \Box_1 \neg A$,
  d.h. Kind 1 weiß in Zustand $A\,\overline{B}\,\overline{C}$, ob es ein schmutziges Gesicht hat.
- $A\,\overline{B}\,\overline{C} \models_K \Box_1 A$,
  d.h. Kind 1 weiß in Zustand $A\,\overline{B}\,\overline{C}$, dass es ein schmutziges Gesicht hat.
- $\overline{A}\,B\,\overline{C} \models_K \Box_2 B$,
  d.h. Kind 2 weiß in Zustand $\overline{A}\,B\,\overline{C}$, dass es ein schmutziges Gesicht hat.
- $\overline{A}\,\overline{B}\,C \models_K \Box_3 C$,
  d.h. Kind 3 weiß in Zustand $\overline{A}\,\overline{B}\,C$, dass es ein schmutziges Gesicht hat.
- $A\,\overline{B}\,C \models_K \Diamond_3 \neg(\Box_1 A \vee \Box_1 \neg A)$
- $A\,B\,C \models_K \Box_3 \neg(\Box_1 A \vee \Box_1 \neg A)$

Das Wissen der Kinder vor der ersten Frage

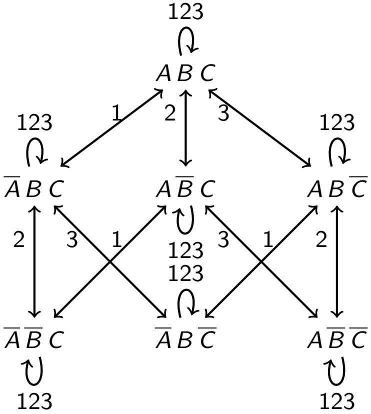
„Mindestens ein Kind hat ein schmutziges Gesicht“



- Zustand $\overline{A}\,\overline{B}\,\overline{C}$ ist nicht möglich.
- $A\,\overline{B}\,\overline{C} \models_K \Box_1 A \vee \Box_1 \neg A$,
  d.h. Kind 1 weiß in Zustand $A\,\overline{B}\,\overline{C}$, ob es ein schmutziges Gesicht hat.
- $A\,\overline{B}\,\overline{C} \models_K \Box_1 A$,
  d.h. Kind 1 weiß in Zustand $A\,\overline{B}\,\overline{C}$, dass es ein schmutziges Gesicht hat.
- $\overline{A}\,B\,\overline{C} \models_K \Box_2 B$,
  d.h. Kind 2 weiß in Zustand $\overline{A}\,B\,\overline{C}$, dass es ein schmutziges Gesicht hat.
- $\overline{A}\,\overline{B}\,C \models_K \Box_3 C$,
  d.h. Kind 3 weiß in Zustand $\overline{A}\,\overline{B}\,C$, dass es ein schmutziges Gesicht hat.
- $A\,\overline{B}\,C \models_K \Diamond_3 \neg(\Box_1 A \vee \Box_1 \neg A)$
- $A\,B\,C \models_K \Box_3 \neg(\Box_1 A \vee \Box_1 \neg A)$

Das Wissen der Kinder vor der ersten Frage

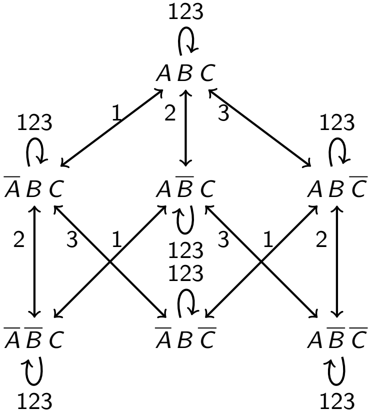
„Mindestens ein Kind hat ein schmutziges Gesicht“



- Zustand $\overline{A}\,\overline{B}\,\overline{C}$ ist nicht möglich.
- $A\,\overline{B}\,\overline{C} \models_K \Box_1 A \vee \Box_1 \neg A$,
  d.h. Kind 1 weiß in Zustand $A\,\overline{B}\,\overline{C}$, ob es ein schmutziges Gesicht hat.
- $A\,\overline{B}\,\overline{C} \models_K \Box_1 A$,
  d.h. Kind 1 weiß in Zustand $A\,\overline{B}\,\overline{C}$, dass es ein schmutziges Gesicht hat.
- $\overline{A}\,B\,\overline{C} \models_K \Box_2 B$,
  d.h. Kind 2 weiß in Zustand $\overline{A}\,B\,\overline{C}$, dass es ein schmutziges Gesicht hat.
- $\overline{A}\,\overline{B}\,C \models_K \Box_3 C$,
  d.h. Kind 3 weiß in Zustand $\overline{A}\,\overline{B}\,C$, dass es ein schmutziges Gesicht hat.
- $A\,\overline{B}\,C \models_K \Diamond_3 \neg(\Box_1 A \vee \Box_1 \neg A)$
- $A\,B\,C \models_K \Box_3 \neg(\Box_1 A \vee \Box_1 \neg A)$

Das Wissen der Kinder vor der ersten Frage

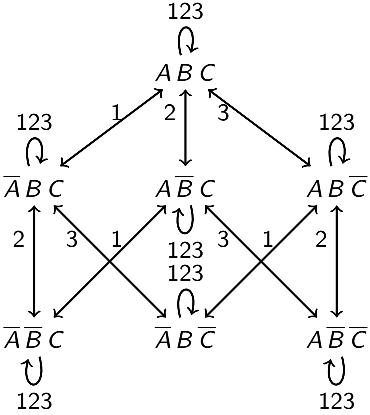
„Mindestens ein Kind hat ein schmutziges Gesicht“



- Zustand $\overline{A}\,\overline{B}\,\overline{C}$ ist nicht möglich.
- $A\,\overline{B}\,\overline{C} \models_K \Box_1 A \vee \Box_1 \neg A$,
  d.h. Kind 1 weiß in Zustand $A\,\overline{B}\,\overline{C}$, ob es ein schmutziges Gesicht hat.
- $A\,\overline{B}\,\overline{C} \models_K \Box_1 A$,
  d.h. Kind 1 weiß in Zustand $A\,\overline{B}\,\overline{C}$, dass es ein schmutziges Gesicht hat.
- $\overline{A}\,B\,\overline{C} \models_K \Box_2 B$,
  d.h. Kind 2 weiß in Zustand $\overline{A}\,B\,\overline{C}$, dass es ein schmutziges Gesicht hat.
- $\overline{A}\,\overline{B}\,C \models_K \Box_3 C$,
  d.h. Kind 3 weiß in Zustand $\overline{A}\,\overline{B}\,C$, dass es ein schmutziges Gesicht hat.
- $A\,\overline{B}\,C \models_K \Diamond_3 \neg(\Box_1 A \vee \Box_1 \neg A)$
- $A\,B\,C \models_K \Box_3 \neg(\Box_1 A \vee \Box_1 \neg A)$

Das Wissen der Kinder vor der ersten Frage

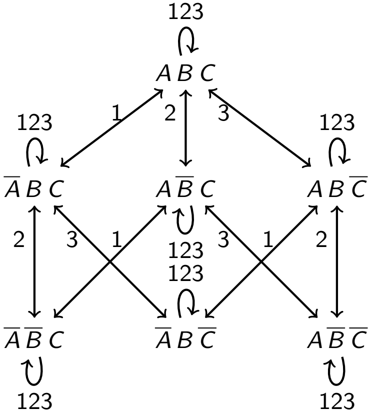
„Mindestens ein Kind hat ein schmutziges Gesicht“



- Zustand $\overline{A}\,\overline{B}\,\overline{C}$ ist nicht möglich.
- $A\,\overline{B}\,\overline{C} \models_K \Box_1 A \vee \Box_1 \neg A$,
  d.h. Kind 1 weiß in Zustand $A\,\overline{B}\,\overline{C}$, ob es ein schmutziges Gesicht hat.
- $A\,\overline{B}\,\overline{C} \models_K \Box_1 A$,
  d.h. Kind 1 weiß in Zustand $A\,\overline{B}\,\overline{C}$, dass es ein schmutziges Gesicht hat.
- $\overline{A}\,B\,\overline{C} \models_K \Box_2 B$,
  d.h. Kind 2 weiß in Zustand $\overline{A}\,B\,\overline{C}$, dass es ein schmutziges Gesicht hat.
- $\overline{A}\,\overline{B}\,C \models_K \Box_3 C$,
  d.h. Kind 3 weiß in Zustand $\overline{A}\,\overline{B}\,C$, dass es ein schmutziges Gesicht hat.
- $A\,\overline{B}\,C \models_K \Diamond_3 \neg(\Box_1 A \vee \Box_1 \neg A)$
- $A\,B\,C \models_K \Box_3 \neg(\Box_1 A \vee \Box_1 \neg A)$

Das Wissen der Kinder nach der ersten Frage

„Wer weiß, ob er ein schmutziges Gesicht hat?“



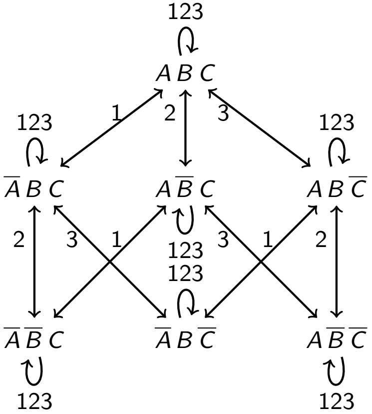
- Niemand meldet sich.
- Also sind die Zustände

  $$\overline{A}\,\overline{B}\,C \quad \overline{A}\,B\,\overline{C} \quad A\,\overline{B}\,\overline{C},$$

  in denen ein Kind weiß, ob es ein schmutziges Gesicht hat, nicht möglich.

Das Wissen der Kinder nach der ersten Frage

„Wer weiß, ob er ein schmutziges Gesicht hat?“



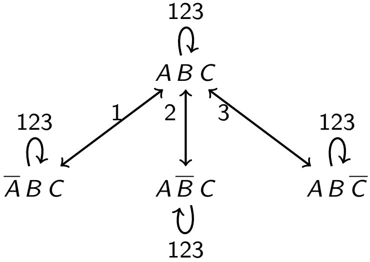
- Niemand meldet sich.
- Also sind die Zustände

  $$\overline{A}\,\overline{B}\,C \quad \overline{A}\,B\,\overline{C} \quad A\,\overline{B}\,\overline{C},$$

  in denen ein Kind weiß, ob es ein schmutziges Gesicht hat, nicht möglich.

Das Wissen der Kinder nach der ersten Frage

„Wer weiß, ob er ein schmutziges Gesicht hat?“



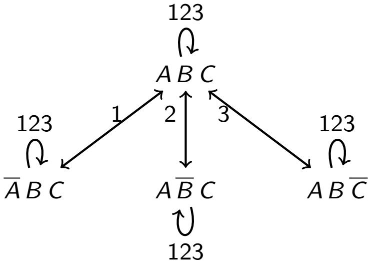
- Niemand meldet sich.
- Also sind die Zustände

$$\overline{A}\,\overline{B}\,C \quad \overline{A}\,B\,\overline{C} \quad A\,\overline{B}\,\overline{C},$$

  in denen ein Kind weiß, ob es ein schmutziges Gesicht hat, nicht möglich.

Das Wissen der Kinder nach der zweiten Frage

„Wer weiß, ob er ein schmutziges Gesicht hat?“



- Alle Zustände erfüllen $\neg B \to (\Box_1 A \wedge \Box_3 C)$.
- Da Kinder 1 und 3 $\neg B$ feststellen, wissen sie, dass sie beide schmutzige Gesichter haben und melden sich.
- Alle Zustände erfüllen

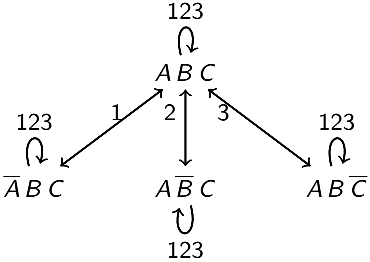
  $$((\Box_1 A \vee \Box_1 \neg A) \wedge (\Box_3 C \vee \Box_3 \neg C)) \to \neg B$$

  „Wenn Kind 1 und Kind 3 wissen, ob sie schmutzige Gesichter haben, dann hat Kind 2 kein schmutziges Gesicht.“

  Also meldet sich Kind 2 auch.

Das Wissen der Kinder nach der zweiten Frage

„Wer weiß, ob er ein schmutziges Gesicht hat?"



- Alle Zustände erfüllen $\neg B \rightarrow (\Box_1 A \wedge \Box_3 C)$.
- Da Kinder 1 und 3 $\neg B$ feststellen, wissen sie, dass sie beide schmutzige Gesichter haben und melden sich.
- Alle Zustände erfüllen

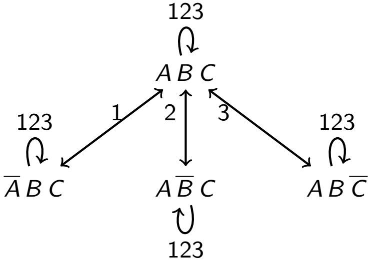
$$((\Box_1 A \vee \Box_1 \neg A) \wedge (\Box_3 C \vee \Box_3 \neg C)) \rightarrow \neg B$$

„Wenn Kind 1 und Kind 3 wissen, ob sie schmutzige Gesichter haben, dann hat Kind 2 kein schmutziges Gesicht."

Also meldet sich Kind 2 auch.

Das Wissen der Kinder nach der dritten Frage

„Wer weiß, ob er ein schmutziges Gesicht hat?“



- Alle Zustände erfüllen $\neg B \to (\Box_1 A \wedge \Box_3 C)$.
- Da Kinder 1 und 3 $\neg B$ feststellen, wissen sie, dass sie beide schmutzige Gesichter haben und melden sich.
- Alle Zustände erfüllen

  $$((\Box_1 A \vee \Box_1 \neg A) \wedge (\Box_3 C \vee \Box_3 \neg C)) \to \neg B$$

  „Wenn Kind 1 und Kind 3 wissen, ob sie schmutzige Gesichter haben,
  dann hat Kind 2 kein schmutziges Gesicht.“

  Also meldet sich Kind 2 auch.

Welche Arten von Wissen gibt es?

- Was man weiß, stimmt tatsächlich.
  $\Box\alpha \rightarrow \alpha$
- Wenn man etwas weiß, dann weiß man auch, dass man es weiß.
  $\Box\alpha \rightarrow \Box\Box\alpha$

Graphen (V, E), die $\Box\alpha \rightarrow \alpha$ für alle α erfüllen,
(d.h. für alle Belegungen ξ gilt $(V, E, \xi) \models_K \Box\alpha \rightarrow \alpha$)
sind reflexiv.

Graphen, die $\Box\alpha \rightarrow \Box\Box\alpha$ für alle α erfüllen,
sind transitiv.

Es gibt also ein „Zusammenspiel“ von
Eigenschaften, die man modellieren will,
Graph-Eigenschaften von Kripke-Modellen, und
Axiomen.

Graph-Eigenschaften für Graphen $G = (V, E)$

reflexiv	$\forall x \in V : (x, x) \in E$
symmetrisch	$\forall x, y \in V : (x, y) \in E \Rightarrow (y, x) \in E$
transitiv	$\forall x, y, z \in V : ((x, y) \in E \;\&\; (y, z) \in E) \Rightarrow (x, z) \in E$
seriell	$\forall x \in V \exists y \in V : (x, y) \in E$
euklidisch	$\forall x, y, z \in V : ((x, y) \in E \;\&\; (x, z) \in E) \Rightarrow (y, z) \in E$
schwach euklidisch	$\forall x, y, z \in V : ((x, y) \in E \;\&\; (x, z) \in E) \Rightarrow$ $((y, z) \in E \text{ oder } (z, y) \in E \text{ oder } y = z)$
schwach gerichtet	$\forall x, y, z \in V : ((x, y) \in E \;\&\; (x, z) \in E \;\&\; y \neq z) \Rightarrow$ $\exists w \in V((y, w) \in E \;\&\; (z, w) \in E))$

schwach funktional	$\forall x, y, z \in V : ((x, y) \in E \;\&\; (x, z) \in E) \Rightarrow y = z$
funktional	schwach funktional und seriell
dicht	$\forall x, y \in V : (x, y) \in E \Rightarrow (\exists z \in V : (x, z) \in E \;\&\; (z, y) \in E)$
konfluent	$\forall x, y \in V : (x, y) \in E \Rightarrow (\exists z \in V : (x, z) \in E \;\&\; (y, z) \in E)$

Axiome für Graph-Eigenschaften von Kripke-Modellen

reflexiv	$\Box\alpha \rightarrow \alpha$	T
symmetrisch	$\alpha \rightarrow \Box\Diamond\alpha$	B
transitiv	$\Box\alpha \rightarrow \Box\Box\alpha$	4
seriell	$\Box\alpha \rightarrow \Diamond\alpha$	D
euklidisch	$\Diamond\alpha \rightarrow \Box\Diamond\alpha$	5
schwach euklidisch	$\Box((\alpha \wedge \Box\alpha) \rightarrow \beta) \vee \Box((\beta \wedge \Box\beta) \rightarrow \alpha)$	.3
schwach gerichtet	$\Diamond(\alpha \wedge \Box\beta) \rightarrow \Box(\alpha \vee \Diamond\beta)$	.2

schwach funktional	$\Diamond\alpha \rightarrow \Box\alpha$
funktional	$\Diamond\alpha \leftrightarrow \Box\alpha$
dicht	$\Box\Box\alpha \rightarrow \Box\alpha$
konfluent	$\Diamond\Box\alpha \rightarrow \Diamond\alpha$

Modallogiken mit bestimmten Graph-Eigenschaften

Die folgenden Logiken sind Erweiterungen der Modallogik K.
Man erhält sie dadurch, dass man
nur noch Kripke-Modelle mit bestimmten Graph-Eigenschaften betrachtet
oder
Axiome zum Frege-Kalkül dazunimmt.

Modallogik	Graph-Eigenschaft	Axiome
K	keine	K
K4	transitiv	K + 4
T	reflexiv	K + T
S4	reflexiv + transitiv	K + T + 4
S4.2	S4 + max.Element	K + T + 4 + .2
S4.3	lineare Ordnung	K + T + 4 + .3
S5	reflexiv und euklidisch	K + T + 5

Schreibweise:
$\models_{S4} \alpha$ gdw. für alle reflexiven und transitiven Graphen G gilt $G \models_{K} \alpha$
gdw. für alle Kripke-Modelle $\mathcal{M}$ gilt: wenn $\mathcal{M} \models_{K} \mathrm{K} \wedge \mathrm{T} \wedge 4$, dann $\mathcal{M} \models_{K} \alpha$

Wissenslogiken

Eigenschaften von Wissen:

K	$\Box(\alpha \to \beta) \to (\Box\alpha \to \Box\beta)$	Verbreitung von Wissen
T	$\Box\alpha \to \alpha$	Korrektheit des Wissens
(Dual	$\Box\alpha \to \neg\Box\neg\alpha$	Konsistenz)
.4	$\Box\alpha \to \Box\Box\alpha$	positive Introspektion
.5	$\Diamond\alpha \to \Box\Diamond\alpha$	negative Introspektion

Epistemische Logiken:

Logiken daraus mit T: *Wissenslogiken*
Logiken daraus ohne T: *Glaubenslogiken*

Andere Logiken mit Kripke-Semantik

Ein Frege-Kalkül für Aussagenlogik mit $\wedge$, $\vee$, $\rightarrow$ und $\perp$ besteht aus folgenden Axiomen und modus ponens.

1. $\alpha \rightarrow (\beta \rightarrow \alpha)$
2. $(\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))$
3. $(\alpha \wedge \beta) \rightarrow \alpha$ und $(\alpha \wedge \beta) \rightarrow \beta$
4. $\alpha \rightarrow (\beta \rightarrow (\alpha \wedge \beta))$
5. $\alpha \rightarrow (\alpha \vee \beta)$ und $\beta \rightarrow (\alpha \vee \beta)$
6. $(\alpha \rightarrow \gamma) \rightarrow ((\beta \rightarrow \gamma) \rightarrow ((\alpha \vee \beta) \rightarrow \gamma))$
7. $(\alpha \rightarrow \beta) \rightarrow ((\alpha \rightarrow \neg\beta) \rightarrow \neg\alpha)$
8. $\neg\neg\alpha \rightarrow \alpha$

Andere Logiken mit Kripke-Semantik

Ein Frege-Kalkül für Aussagenlogik mit $\wedge$, $\vee$, $\rightarrow$ und $\perp$ besteht aus folgenden Axiomen und modus ponens.

1. $\alpha \rightarrow (\beta \rightarrow \alpha)$
2. $(\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))$
3. $(\alpha \wedge \beta) \rightarrow \alpha$ und $(\alpha \wedge \beta) \rightarrow \beta$
4. $\alpha \rightarrow (\beta \rightarrow (\alpha \wedge \beta))$
5. $\alpha \rightarrow (\alpha \vee \beta)$ und $\beta \rightarrow (\alpha \vee \beta)$
6. $(\alpha \rightarrow \gamma) \rightarrow ((\beta \rightarrow \gamma) \rightarrow ((\alpha \vee \beta) \rightarrow \gamma))$
7. $(\alpha \rightarrow \beta) \rightarrow ((\alpha \rightarrow \neg\beta) \rightarrow \neg\alpha)$
8. $\neg\neg\alpha \rightarrow \alpha$

Durch Weglassen des Axioms 8 (Doppelnegationsgesetz) erhält man einen Kalkül für die *intuitionistische Logik*.

Wie sieht eine Semantik für die intuitionistische Logik aus?

Bem.: nicht-intuitionistische Beweise

Mathematische Beweise können nicht-konstruktiv sein.

Beispiel-Satz

Es gibt nicht-rationale Zahlen a und b, so dass a^b rational ist.

Beweis:

Wir wissen bereits: $\sqrt{2}$ ist eine nicht-rationale Zahl.

Fall 1: ${\sqrt{2}}^{\sqrt{2}}$ ist rational.
Dann kann $a = b = \sqrt{2}$ gewählt werden.
$a^b = {\sqrt{2}}^{\sqrt{2}}$ ist rational (laut Voraussetzung).

Fall 2: ${\sqrt{2}}^{\sqrt{2}}$ ist nicht rational.
Dann kann $a = {\sqrt{2}}^{\sqrt{2}}$ und $b = \sqrt{2}$ gewählt werden.
$a^b = ({\sqrt{2}}^{\sqrt{2}})^{\sqrt{2}} = {\sqrt{2}}^{\sqrt{2}\cdot\sqrt{2}} = {\sqrt{2}}^{\,2} = 2$ ist rational. ✓

Obwohl der Satz bewiesen ist, ist nicht klar, welche der beiden Wahlmöglichkeiten für a und b die gewünschte Eigenschaft hat.

Intuitionistische Logik (intuitiv)

Klassische Aussagenlogik (Frege):

- eine Aussage kann wahr (oder falsch) sein;
  der Wahrheitswert zusammengesetzter Aussagen wird aus den Wahrheitswerten ihrer Teile bestimmt

Intuitionistische Aussagenlogik (Brouwer):

- für eine Aussage können wir einen Beweis kennen (oder nicht);
  aus einem Beweis für eine zusammengesetzte Aussage können wir Beweise für ihre Teile bilden

Am Beispiel:
„$\mathrm{P} = \mathrm{NP}$ oder $\mathrm{P} \neq \mathrm{NP}$“ ist klassisch *wahr*
aber intuitionistisch *falsch* (im Augenblick)

Formeln, die intuitionistisch nicht gültig sind

Beispiele:

- $A \vee \neg A$ (allgemeiner: Gesetz des ausgeschlossenen Dritten)
- $\neg A \vee \neg\neg A$ (schwaches Gesetz des ausgeschlossenen Dritten)
- $\neg\neg A \rightarrow A$ (Doppelnegationsgesetz)
- $(A \rightarrow B) \vee (B \rightarrow A)$
- $\neg(A \wedge B) \rightarrow (\neg A \vee \neg B)$ (de Morgansche Regel)
- $(A \rightarrow B) \rightarrow (\neg A \vee B)$

Intuitionistische Aussagenlogik formal

Definition 10.1 (Formeln der intuitionistischen Aussagenlogik)

Formeln der intuitionistischen Aussagenlogik sind induktiv definiert wie folgt.

1. Die Konstante $\perp$ und alle atomaren Formeln sind Formeln.
2. Für alle Formeln α und β
   sind $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$ und $(\alpha \rightarrow \beta)$ ebenfalls Formeln.

(3. Es gibt keine anderen Formeln.)

$\neg\alpha$ ist abkürzende Schreibweise für $\alpha \rightarrow \perp$.

$\top$ ist abkürzende Schreibweise für $\perp \rightarrow \perp$.

Definition 10.2 (Semantik der intuitionistischen Aussagenlogik)

Ein intuitionistisches Kripke-Modell $\mathcal{M} = (W, \leq, \xi)$ besteht aus

- einem Graph $(W, \leq)$
  mit Welten $W \neq \emptyset$, und reflexiver und transitiver Kantenrelation $\leq$, und
- einer Belegungsfunktion $\xi : W \to$ Atom-Mengen mit
  $u \leq w \Rightarrow \xi(u) \subseteq \xi(w)$ für alle $u, w \in W$ (*Persistenz*).

Die Relation $\models_{i}$ ist für intuitionistische Kripke-Modelle $\mathcal{M} = (W, \leq, \xi)$, $w \in W$ und Formeln α und β sowie Atome A_i wie folgt definiert:

$$
\begin{array}{lll}
\mathcal{M}, w \not\models_{i} \bot & & \\
\mathcal{M}, w \models_{i} A_i & \text{gdw.} & A_i \in \xi(w) \\
\mathcal{M}, w \models_{i} \alpha \wedge \beta & \text{gdw.} & \mathcal{M}, w \models_{i} \alpha \text{ und } \mathcal{M}, w \models_{i} \beta \\
\mathcal{M}, w \models_{i} \alpha \vee \beta & \text{gdw.} & \mathcal{M}, w \models_{i} \alpha \text{ oder } \mathcal{M}, w \models_{i} \beta \\
\mathcal{M}, w \models_{i} \alpha \to \beta & \text{gdw.} & \text{für alle } t \in W \text{ mit } w \leq t\text{:} \\
 & & \qquad\qquad \text{wenn } \mathcal{M}, t \models_{i} \alpha, \text{ dann } \mathcal{M}, t \models_{i} \beta
\end{array}
$$

Lemma 10.3 (Persistenz gilt für alle Formeln)

Sei $\mathcal{M} = (W, \leq, \xi)$ *ein intuitionistisches Kripke-Modell.*
Dann gilt für alle intuitionistischen Formeln α *und* $u, v \in W$*:*

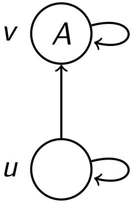
1. *Wenn* $\mathcal{M}, u \models_{\mathsf{i}} \alpha$ *und* $u \leq v$*, dann* $\mathcal{M}, v \models_{\mathsf{i}} \alpha$*.*
2. *Wenn* $\mathcal{M}, v \not\models_{\mathsf{i}} \alpha$ *und* $u \leq v$*, dann* $\mathcal{M}, u \not\models_{\mathsf{i}} \alpha$*.*

Definition 10.4 (Gültigkeit)

Sei α eine intuitionistische Formel.
α heißt *(intuitionistisch) gültig* ($\models_{\mathsf{i}} \alpha$), falls $\mathcal{M}, w \models_{\mathsf{i}} \alpha$ für jedes intuitionistische Modell $\mathcal{M}$ und jede Welt w von $\mathcal{M}$ gilt.

$\not\models_{\mathrm{i}} A \vee \neg A$

Das Gesetz des ausgeschlossenen Dritten ist intuitionistisch nicht gültig



$\mathcal{M}_1$

$\mathcal{M}_1, v \models_{\mathrm{i}} A,\ \ A \vee \neg A$

$\mathcal{M}_1, v \not\models_{\mathrm{i}} \ \neg A$

$\mathcal{M}_1, u \not\models_{\mathrm{i}} A,\ \ \neg A,\ \ A \vee \neg A$

$\mathcal{M}_1, u \models_{\mathrm{i}} A \vee \neg A$

gdw. $\mathcal{M}_1, u \models_{\mathrm{i}} A$ oder $\mathcal{M}_1, u \models_{\mathrm{i}} \neg A$

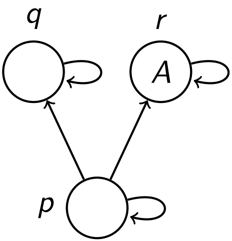
gdw. $\mathcal{M}_1, u \models_{\mathrm{i}} \neg A$

gdw. für alle t mit $u \leq t$: wenn $\mathcal{M}_1, t \models_{\mathrm{i}} A$, dann $\mathcal{M}_1, t \models_{\mathrm{i}} \perp$

falsche Aussage: $u \leq v$ und $\mathcal{M}_1, v \models_{\mathrm{i}} A$ und $\mathcal{M}_1, v \not\models_{\mathrm{i}} \perp$

$\not\models_{i} \neg(A \wedge \neg A) \rightarrow (\neg A \vee \neg\neg A)$

Eine deMorgan-Regel ist intuitionistisch nicht gültig



$\mathcal{M}_2$

$$\mathcal{M}_2, r \models_{i} A,\ \neg\neg A, \neg(A \wedge \neg A)$$
$$\mathcal{M}_2, r \not\models_{i} \neg A$$

$$\mathcal{M}_2, q \models_{i} \neg A, \neg(A \wedge \neg A)$$
$$\mathcal{M}_2, q \not\models_{i} A,\ \neg\neg A$$

$$\mathcal{M}_2, p \models_{i} \neg(A \wedge \neg A)$$
$$\mathcal{M}_2, p \not\models_{i} A,\ \neg A,\ \neg\neg A,\ \neg A \vee \neg\neg A,\ \neg(A \wedge \neg A) \rightarrow (\neg A \vee \neg\neg A)$$

Klassische vs. intuitionistische Aussagenlogik

Satz 10.5 (Satz von Glivenko)

Für jede (intuitionistische/aussagenlogische) Formel α gilt:

$\models \alpha$ *genau dann, wenn* $\models_{\mathrm{i}} \neg\neg\alpha$.

Welche Formeln, die nicht mit $\neg\neg$ anfangen, sind intuitionistisch gültig?

Modale vs. intuitionistische Aussagenlogik

Definition 10.6 (Gödels Übersetzung intuitionistischer Formeln)

Die Abbildung g : *intuitionistische Formeln* $\to$ *modale Formeln* ist wie folgt induktiv definiert.

$$g(\alpha) = \begin{cases} \bot, & \text{falls } \alpha = \bot \\ \Box A_i, & \text{falls } \alpha = A_i \text{ ein Atom ist} \\ g(\beta) \wedge g(\gamma), & \text{falls } \alpha = \beta \wedge \gamma \text{ eine Konjunktion ist} \\ g(\beta) \vee g(\gamma), & \text{falls } \alpha = \beta \vee \gamma \text{ eine Disjunktion ist} \\ \Box(g(\beta) \to g(\gamma)), & \text{falls } \alpha = \beta \to \gamma \text{ eine Implikation ist} \end{cases}$$

Bsp.: $g((A \wedge B) \to \neg(A \vee B)) = \Box((\Box A \wedge \Box B) \to \Box\neg(\Box A \vee \Box B))$.

Satz 10.7 (McKinsey, Tarski)

Für jede intuitionistische Formel α gilt

$$\models_{\mathrm{i}} \alpha \text{ genau dann, wenn } \models_{S4} g(\alpha).$$

Mit der Gödel-Übersetzung erhält man
modale Begleiter zu intuitionistischen Logiken

modaler Begleiter	int. Logik	
K4	BPL	IPC mit schwachem MP (basic propositional logic)
S4	IPL	intuitionistic propositional logic
S4.2	KC	IPL $+ \neg\neg\neg\alpha \to \neg\alpha$
S4.3	LC	IPL $+ (\alpha \to \beta) \vee (\beta \to \alpha)$
S5	AL	IPL $+ \neg\neg\alpha \to \alpha$

Was haben wir in Teil 2 gelernt?

- Wir kennen modallogische Formeln und ihre Semantik mit Kripke-Modellen.
- Wir wissen, dass die Axiome des Frege-Kalküls gültig sind und können den Beweis für Axiom (K) reproduzieren.
- Wir wissen, dass im Frege-Kalkül nur gültige Formeln herleitbar sind und können den Beweis reproduzieren (Indukt. über Länge der Herleit.).
- Wir wissen, dass jede gültige Formel im Tableau-Kalkül bewiesen werden kann und können den Beweis reproduzieren (Induktion über die Pfadlänge in geschlossenen Tableaux).
- Wir wissen, wie man Tableau-Beweise in Frege-Beweise umwandeln kann und können den Beweis reproduzieren (Induktion über die Pfadlänge in widersprüchlichen Tableaux).
- Wir wissen, wie die Vollständigkeit des Frege-Kalküls aus der Vollständigkeit des Tableau-Kalküls gefolgert werden kann.
- Wir wissen, wie die Korrektheit des Tableau-Kalküls aus der Korrektheit des Frege-Kalküls gefolgert werden kann.
- ... das alles ist nur der Anfang ... es gibt noch viel mehr Logiken ...

Teil 3: Temporale Aussagenlogik

1. Aussagenlogik

2. Modale Aussagenlogik

3. Temporale Aussagenlogik

Temporale Logik benutzt man z.B. zur Untersuchung von Prozessen, die schrittweise ihren Zustand verändern.

Eine Aussage kann z.B.

- jetzt, oder
- irgendwann in der Zukunft, oder
- solange, bis eine andere Aussage wahr ist,

wahr sein.

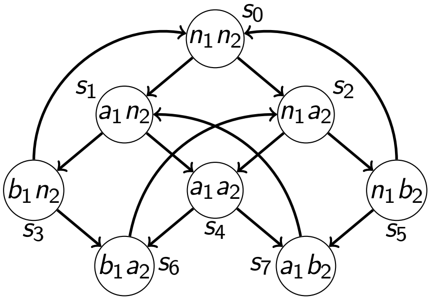
Einführendes Beispiel für Linear Time Logic LTL

Modelliere: in einem System laufen mehrere Prozesse,
die nicht gleichzeitig eine bestimmte Ressource benutzen dürfen.
Prozess i ist in einem der drei Zustände

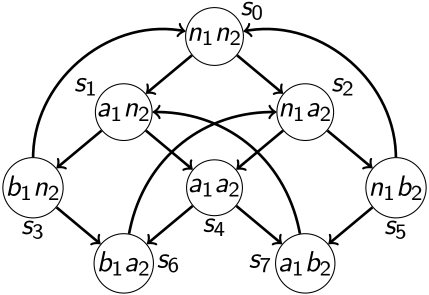
- n_i – Ressource wird von Prozess i weder angefordert noch benutzt
- a_i – Ressource wird von Prozess i angefordert (es wird auf Zugriff gewartet)
- b_i – Ressource wird von Prozess i benutzt

Modell für erlaubte Zustandsübergänge
bei 2 Prozessen:

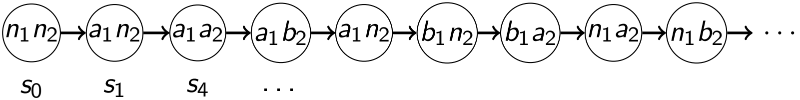
(wir schreiben Atome hier mit kleinen Buchst.)



Teste, ob dieses Modell garantiert, dass
jeder Prozess die Ressource – nachdem er sie angefordert hat – auch irgendwann benutzen kann!



Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:



Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$\begin{array}{cccccccccccccccccccc}
n_1 n_2 & \to & a_1 n_2 & \to & a_1 a_2 & \to & a_1 b_2 & \to & a_1 n_2 & \to & b_1 n_2 & \to & b_1 a_2 & \to & n_1 a_2 & \to & n_1 b_2 & \to & \cdots \\
s_0 & & s_1 & & s_4 & & \ldots & & & & & & & & & & & &
\end{array}$$

Eigenschaften von Pfad π:

$\pi \models_p n_1 \wedge n_2$ „die Startwelt von π erfüllt $n_1 \wedge n_2$"

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$n_1 n_2 \to a_1 n_2 \to a_1 a_2 \to a_1 b_2 \to a_1 n_2 \to b_1 n_2 \to b_1 a_2 \to n_1 a_2 \to n_1 b_2 \to \cdots$$

$$s_0 \quad s_1 \quad s_4 \quad \ldots$$

Eigenschaften von Pfad π:

$\pi \models_p \mathsf{F}\, b_2$ „π erreicht irgendwann einen Zustand, der b_2 erfüllt“

F ... future

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$\underset{s_0}{n_1\,n_2} \rightarrow \underset{s_1}{a_1\,n_2} \rightarrow \underset{s_4}{a_1\,a_2} \rightarrow \underset{\ldots}{a_1\,b_2} \rightarrow a_1\,n_2 \rightarrow b_1\,n_2 \rightarrow b_1\,a_2 \rightarrow n_1\,a_2 \rightarrow n_1\,b_2 \rightarrow \cdots$$

Eigenschaften von Pfad π:

$\pi \models_p \mathsf{F}\, b_2$	„π erreicht irgendwann einen Zustand, der b_2 erfüllt“
	„es gibt einen Suffix ρ von $\pi = \pi'\rho$ mit $\rho \models_p b_2$“

F . . . future

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$\begin{array}{ccccccccccccccccccc}
n_1 n_2 & \to & a_1 n_2 & \to & a_1 a_2 & \to & a_1 b_2 & \to & a_1 n_2 & \to & b_1 n_2 & \to & b_1 a_2 & \to & n_1 a_2 & \to & n_1 b_2 & \to & \cdots \\
s_0 & & s_1 & & s_4 & & \ldots & & & & & & & & & & & &
\end{array}$$

Eigenschaften von Pfad π:

$\pi \models_{p} \mathsf{G}\ \neg(b_1 \wedge b_2)$ „überall auf π ist $\neg(b_1 \wedge b_2)$ erfüllt“

F … future, G … globally

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$\begin{array}{cccccccccc} n_1 n_2 \to & a_1 n_2 \to & a_1 a_2 \to & a_1 b_2 \to & a_1 n_2 \to & b_1 n_2 \to & b_1 a_2 \to & n_1 a_2 \to & n_1 b_2 \to & \cdots \\ s_0 & s_1 & s_4 & \ldots & & & & & & \end{array}$$

Eigenschaften von Pfad π:

$\pi \models_p \mathsf{G}\ \neg(b_1 \wedge b_2)$ „überall auf π ist $\neg(b_1 \wedge b_2)$ erfüllt“

„für jeden Suffix ρ von $\pi = \sigma\rho$ gilt $\rho \models_p \neg(b_1 \wedge b_2)$“

F ... future, G ... globally

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$\begin{array}{cccccccccccccccccc}
n_1\,n_2 & \rightarrow & a_1\,n_2 & \rightarrow & a_1\,a_2 & \rightarrow & a_1\,b_2 & \rightarrow & a_1\,n_2 & \rightarrow & b_1\,n_2 & \rightarrow & b_1\,a_2 & \rightarrow & n_1\,a_2 & \rightarrow & n_1\,b_2 & \rightarrow \cdots \\
s_0 & & s_1 & & s_4 & & \ldots & & & & & & & & & & &
\end{array}$$

Eigenschaften von Pfad π:

$\pi \models_p \mathsf{G}(a_2 \rightarrow \mathsf{F}\, b_2)$ „überall auf π ist erfüllt:

wenn a_2 erfüllt ist, dann wird irgendwann b_2 erfüllt“

F ... future, G ... globally

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$\begin{array}{cccccccccccccccccc}
n_1\,n_2 & \to & a_1\,n_2 & \to & a_1\,a_2 & \to & a_1\,b_2 & \to & a_1\,n_2 & \to & b_1\,n_2 & \to & b_1\,a_2 & \to & n_1\,a_2 & \to & n_1\,b_2 & \to \cdots \\
s_0 & & s_1 & & s_4 & & \ldots & & & & & & & & & & &
\end{array}$$

Eigenschaften von Pfad π:

$\pi \models_p \mathsf{G}(a_2 \to \mathsf{F}\, b_2)$

„überall auf π ist erfüllt:

wenn a_2 erfüllt ist, dann wird irgendwann b_2 erfüllt“

„für jeden Suffix ρ von $\pi = \sigma\rho$ gilt:

wenn $\rho \models_p a_2$, dann $\rho \models_p \mathsf{F}\, b_2$“

F ... future, G ... globally

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$\underset{s_0}{n_1 n_2} \to \underset{s_1}{a_1 n_2} \to \underset{s_4}{a_1 a_2} \to \underset{\ldots}{a_1 b_2} \to a_1 n_2 \to b_1 n_2 \to b_1 a_2 \to n_1 a_2 \to n_1 b_2 \to \cdots$$

Eigenschaften von Pfad π:

$\pi \models_p \mathsf{X}\, n_2$ „π erfüllt im nächsten Zustand n_2“

F ... future, G ... globally, X ... next

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$n_1 n_2 \to a_1 n_2 \to a_1 a_2 \to a_1 b_2 \to a_1 n_2 \to b_1 n_2 \to b_1 a_2 \to n_1 a_2 \to n_1 b_2 \to \cdots$$

$$s_0 \quad s_1 \quad s_4 \quad \ldots$$

Eigenschaften von Pfad π:

$\pi \models_p \mathrm{X}\, n_2$ „für ρ mit $\pi = s\rho$ gilt $\rho \models_p n_2$“

F … future, G … globally, X … next

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$
\begin{array}{ccccccccccccccccccc}
n_1 n_2 & \to & a_1 n_2 & \to & a_1 a_2 & \to & a_1 b_2 & \to & a_1 n_2 & \to & b_1 n_2 & \to & b_1 a_2 & \to & n_1 a_2 & \to & n_1 b_2 & \to & \cdots \\
s_0 & & s_1 & & s_4 & & \ldots & & & & & & & & & & & &
\end{array}
$$

Eigenschaften von Pfad π:

$\pi \models_p n_1 \wedge (\mathrm{X}\, n_2 \wedge \mathrm{X}\,\mathrm{X}\, a_2)$ „π erfüllt am Start n_1, danach n_2 und danach a_2"

F … future, G … globally, X … next

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$(n_1\, n_2) \to (a_1\, n_2) \to (a_1\, a_2) \to (a_1\, b_2) \to (a_1\, n_2) \to (b_1\, n_2) \to (b_1\, a_2) \to (n_1\, a_2) \to (n_1\, b_2) \to \cdots$$

$$s_0 \qquad s_1 \qquad s_4 \qquad \ldots$$

Eigenschaften von Pfad π:

$\pi \models_p n_1 \wedge (\mathrm{X}\, n_2 \wedge \mathrm{X}\,\mathrm{X}\, a_2)$ „π erfüllt am Start n_1, danach n_2 und danach a_2"

„$\pi \models_p n_1$, und für ρ mit $\pi = z\rho$ gilt $\rho \models_p n_2$,
und für τ mit $\rho = s\tau$ gilt $\tau \models_p a_2$"

F ... future, G ... globally, X ... next

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$\underset{s_0}{(n_1 n_2)} \to \underset{s_1}{(a_1 n_2)} \to \underset{s_4}{(a_1 a_2)} \to \underset{\ldots}{(a_1 b_2)} \to (a_1 n_2) \to (b_1 n_2) \to (b_1 a_2) \to (n_1 a_2) \to (n_1 b_2) \to \cdots$$

Eigenschaften von Pfad π:

$\pi \models_p n_1 \wedge (\mathrm{X}\, n_2 \wedge \mathrm{X}\,\mathrm{X}\, a_2)$ „π erfüllt am Start n_1, danach n_2 und danach a_2"

„$\pi \models_p n_1$, und für ρ mit $\pi = z\rho$ gilt $\rho \models_p n_2$,
und für τ mit $\rho = s\tau$ gilt $\tau \models_p a_2$"

„$\pi \models_p n_1$, und für ρ mit $\pi = z\rho$ gilt $\rho \models_p n_2 \wedge \mathrm{X}\, a_2$"

F … future, G … globally, X … next

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$\underset{s_0}{(n_1 n_2)} \to \underset{s_1}{(a_1 n_2)} \to \underset{s_4}{(a_1 a_2)} \to \underset{\ldots}{(a_1 b_2)} \to (a_1 n_2) \to (b_1 n_2) \to (b_1 a_2) \to (n_1 a_2) \to (n_1 b_2) \to \cdots$$

Eigenschaften von Pfad π:

$\pi \models_{p} n_1 \land (\mathsf{X}\, n_2 \land \mathsf{X}\,\mathsf{X}\, a_2)$ „π erfüllt am Start n_1, danach n_2 und danach a_2"

„$\pi \models_{p} n_1$, und für ρ mit $\pi = z\rho$ gilt $\rho \models_{p} n_2$,
und für τ mit $\rho = s\tau$ gilt $\tau \models_{p} a_2$"

„$\pi \models_{p} n_1$, und für ρ mit $\pi = z\rho$ gilt $\rho \models_{p} n_2 \land \mathsf{X}\, a_2$"

$\pi \models_{p} n_1 \land \mathsf{X}(n_2 \land \mathsf{X}\, a_2)$

F … future, G … globally, X … next

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \dots$ durch das Modell:

$$(n_1 n_2) \to (a_1 n_2) \to (a_1 a_2) \to (a_1 b_2) \to (a_1 n_2) \to (b_1 n_2) \to (b_1 a_2) \to (n_1 a_2) \to (n_1 b_2) \to \cdots$$

$$s_0 \qquad s_1 \qquad s_4 \qquad \dots$$

Eigenschaften von Pfad π:

$\pi \models_p \neg\, \mathsf{G}((a_1 \wedge \neg b_2) \to \mathsf{X}\, b_1)$ „nicht überall auf π, wo $a_1 \wedge \neg b_2$ erfüllt wird, wird im nächsten Zustand b_1 erfüllt“

F … future, G … globally, X … next

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$\underset{s_0}{(n_1 n_2)} \to \underset{s_1}{(a_1 n_2)} \to \underset{s_4}{(a_1 a_2)} \to \underset{\ldots}{(a_1 b_2)} \to (a_1 n_2) \to (b_1 n_2) \to (b_1 a_2) \to (n_1 a_2) \to (n_1 b_2) \to \cdots$$

Eigenschaften von Pfad π:

$\pi \models_p \neg\, \mathsf{G}((a_1 \wedge \neg b_2) \to \mathsf{X}\, b_1)$ „nicht überall auf π, wo $a_1 \wedge \neg b_2$ erfüllt wird,
wird im nächsten Zustand b_1 erfüllt“

„irgendwo auf π, wo $a_1 \wedge \neg b_2$ erfüllt wird,
wird nicht im nächsten Zustand b_1 erfüllt“

F . . . future, G . . . globally, X . . . next

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$\begin{array}{ccccccccccccccccccc}
n_1 n_2 & \to & a_1 n_2 & \to & a_1 a_2 & \to & a_1 b_2 & \to & a_1 n_2 & \to & b_1 n_2 & \to & b_1 a_2 & \to & n_1 a_2 & \to & n_1 b_2 & \to & \cdots \\
s_0 & & s_1 & & s_4 & & \ldots & & & & & & & & & & & &
\end{array}$$

Eigenschaften von Pfad π:

$\pi \models_p \neg\, \mathsf{G}((a_1 \wedge \neg b_2) \to \mathsf{X}\, b_1)$ „nicht überall auf π, wo $a_1 \wedge \neg b_2$ erfüllt wird,
wird im nächsten Zustand b_1 erfüllt“

„irgendwo auf π, wo $a_1 \wedge \neg b_2$ erfüllt wird,
wird nicht im nächsten Zustand b_1 erfüllt“

$\pi \models_p \mathsf{F}((a_1 \wedge \neg b_2) \wedge \neg\, \mathsf{X}\, b_1)$

F . . . future, G . . . globally, X . . . next

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$\underset{s_0}{n_1 n_2} \to \underset{s_1}{a_1 n_2} \to \underset{s_4}{a_1 a_2} \to \underset{\ldots}{a_1 b_2} \to a_1 n_2 \to b_1 n_2 \to b_1 a_2 \to n_1 a_2 \to n_1 b_2 \to \cdots$$

Eigenschaften von Pfad π:

$\pi \models_p \neg\, \mathrm{G}((a_1 \wedge \neg b_2) \to \mathrm{X}\, b_1)$ „nicht überall auf π, wo $a_1 \wedge \neg b_2$ erfüllt wird,
wird im nächsten Zustand b_1 erfüllt“

„irgendwo auf π, wo $a_1 \wedge \neg b_2$ erfüllt wird,
wird nicht im nächsten Zustand b_1 erfüllt“

$\pi \models_p \mathrm{F}((a_1 \wedge \neg b_2) \wedge \neg\, \mathrm{X}\, b_1)$

$\pi \models_p \mathrm{F}((a_1 \wedge \neg b_2) \wedge \mathrm{X} \neg b_1)$

F … future, G … globally, X … next

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$n_1 n_2 \to a_1 n_2 \to a_1 a_2 \to a_1 b_2 \to a_1 n_2 \to b_1 n_2 \to b_1 a_2 \to n_1 a_2 \to n_1 b_2 \to \cdots$$

$s_0 \quad s_1 \quad s_4 \quad \ldots$

Eigenschaften von Pfad π:

$\pi \models_p (a_1 \vee n_2) \text{ U } b_1$ „vom Start von π aus wird $a_1 \vee n_2$ solange erfüllt, bis b_1 erfüllt wird“

F ... future, G ... globally, X ... next , U ... until

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$\underset{s_0}{(n_1 n_2)} \to \underset{s_1}{(a_1 n_2)} \to \underset{s_4}{(a_1 a_2)} \to \underset{\ldots}{(a_1 b_2)} \to (a_1 n_2) \to (b_1 n_2) \to (b_1 a_2) \to (n_1 a_2) \to (n_1 b_2) \to \cdots$$

Eigenschaften von Pfad π:

$\pi \models_{p} (a_1 \vee n_2)$ U b_1

„vom Start von π aus wird $a_1 \vee n_2$ solange erfüllt, bis b_1 erfüllt wird"

„es gibt eine Aufteilung von $\pi = \rho\tau$ in einen Präfix ρ und einen Suffix τ, so dass überall auf ρ $a_1 \vee n_2$ erfüllt ist und τ b_1 erfüllt"

F . . . future, G . . . globally, X . . . next , U . . . until

Ein (unendlicher) Pfad $\pi = s_0 s_1 s_4 \ldots$ durch das Modell:

$$n_1 n_2 \to a_1 n_2 \to a_1 a_2 \to a_1 b_2 \to a_1 n_2 \to b_1 n_2 \to b_1 a_2 \to n_1 a_2 \to n_1 b_2 \to \cdots$$

$s_0 \quad s_1 \quad s_4 \quad \ldots$

Eigenschaften von Pfad π:

$\pi \models_p b_2 \; \mathsf{R} \, (n_1 \vee a_1)$ „vom Start von π aus wird $n_1 \vee a_1$ solange erfüllt, bis (möglicherweise) gleichzeitig b_2 erfüllt wird“

F … future, G … globally, X … next , U … until , R … release

Teil 3: Temporale Aussagenlogik

[Literatur (siehe Semesterapparat):

Baier, Katoen: Principles of model checking — VL10

Hoffmann, Lange: Automatentheorie und Logik — VL11/12]

Vorlesung 10:
Grundbegriffe der Zeitlogik LTL und ihrer Kalküle

3. Temporale Aussagenlogik

VL10: Grundbegriffe der Zeitlogik LTL und ihrer Kalküle

- Syntax und Semantik
- Ein Tableau-Kalkül für LTL
- Ein Frege-Kalkül für LTL

VL11: Büchi-Automaten

VL12: Ein Algorithmus für das Gültigkeitsproblem für LTL

Exkurs und Abschluss

[Literatur: Kreuzer, Kühling: Logik für Informatiker.
Baier, Katoen: Principles of model checking.
weitere Literaturhinweise im Text]

Wir werden die Grundbegriffe der *linear time logic* LTL definieren.
Dazu erweitern wir die Begriffe, die wir bereits kennen.

- Wie sehen LTL-Formeln aus?
- Was sind Pfad-Semantik und Kripke-Semantik von LTL-Formeln?
- Welche temporalen Verknüpfungszeichen braucht man? (adäquate Verknüpfungszeichen)
- Wie sieht ein Tableau-Kalkül für LTL aus?
- Wie sieht ein Frege-Kalkül für LTL aus?

10.1 Syntax und Semantik von LTL

LTL ... linear-time temporal logic

Definition 10.1 (die Sprache: LTL-Formeln)

(Aussagenlogische) LTL-Formeln sind induktiv definiert wie folgt.

1. Die Konstante $\perp$ und alle Atome A_i sind LTL-Formeln.
2. Für alle LTL-Formeln α und β sind

   $(\alpha \to \beta)$,
   $(\mathsf{X}\,\alpha)$ (next),
   $(\mathsf{F}\,\alpha)$ (future),
   $(\mathsf{G}\,\alpha)$ (globally),
   $(\alpha\ \mathsf{U}\ \beta)$ (until),
   $(\alpha\ \mathsf{R}\ \beta)$ (release)

   ebenfalls LTL-Formeln.

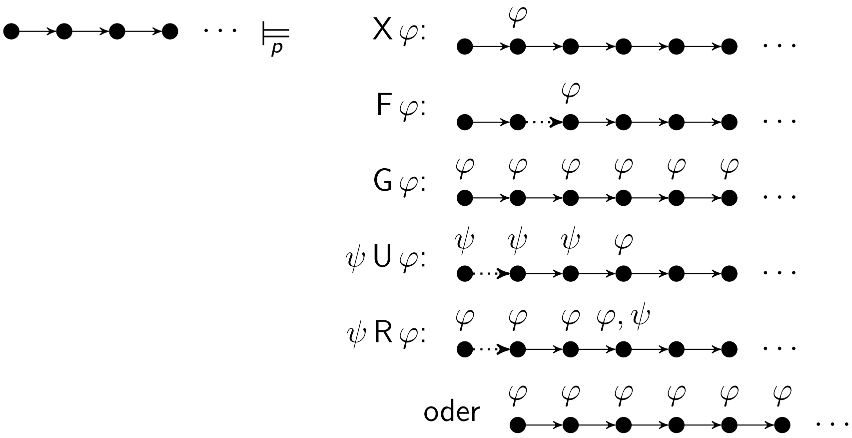
Wir benutzen Abkürzungen $\top$, $\neg\alpha$, $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \leftrightarrow \beta$ wie üblich.
Überflüssige Klammern werden weggelassen.

Definition 10.2 (Pfad-Erfüllungsrelation $\models_p$ für LTL)

Sei $\pi = \pi_0\pi_1\pi_2\ldots$ eine unendliche Folge von Belegungen $\pi_i \subseteq \{A_0, A_1, \ldots\}$, und $\pi^j = \pi_j\pi_{j+1}\ldots$ Suffix von π (für $j \geqslant 0$). Die Relation $\models_p$ zwischen Belegungsfolgen und LTL-Formeln ist wie folgt definiert.

$$
\begin{array}{rll}
\pi \not\models_p \bot & & \\
\pi \models_p A_i & \text{gdw.} & A_i \in \pi_0 \\
\pi \models_p \alpha \to \beta & \text{gdw.} & \pi \not\models_p \alpha \text{ oder } \pi \models_p \beta \\
\pi \models_p \mathsf{X}\,\alpha & \text{gdw.} & \pi^1 \models_p \alpha \\
\pi \models_p \mathsf{F}\,\alpha & \text{gdw.} & \text{es gibt ein } i \geqslant 0 : \pi^i \models_p \alpha \\
\pi \models_p \mathsf{G}\,\alpha & \text{gdw.} & \text{für alle } i \geqslant 0 : \pi^i \models_p \alpha \\
\pi \models_p \alpha\,\mathsf{U}\,\beta & \text{gdw.} & \text{es gibt ein } i \geqslant 0 : \pi^i \models_p \beta \\
& & \qquad \text{und für alle } j < i : \pi^j \models_p \alpha \\
\pi \models_p \alpha\,\mathsf{R}\,\beta & \text{gdw.} & \text{für alle } i \geqslant 0 : \pi^i \models_p \beta, \text{ oder} \\
& & \text{es gibt ein } j \geqslant 0 : \pi^j \models_p \alpha \text{ und für alle } k \leqslant j : \pi^k \models_p \beta
\end{array}
$$

Intuitive Vorstellung von $\models_p$ ...



Definition 10.3 (äquivalente LTL-Formeln)

LTL-Formeln α und β heißen äquivalent ($\alpha \equiv \beta$),
falls für jede Belegungsfolge π gilt: $\pi \models_{p} \alpha$ genau dann, wenn $\pi \models_{p} \beta$.

Lemma 10.4 (wichtige Äquivalenzen in LTL)

1. $\neg \mathsf{X}\, \alpha \equiv \mathsf{X} \neg\alpha$
2. $\neg \mathsf{G}\, \alpha \equiv \mathsf{F} \neg\alpha$
3. $\neg(\alpha \mathbin{\mathsf{R}} \beta) \equiv \neg\alpha \mathbin{\mathsf{U}} \neg\beta$
4. $\mathsf{F}\, \alpha \equiv \top \mathbin{\mathsf{U}} \alpha$
5. $\mathsf{X}\, \alpha \equiv \mathsf{X}(\bot \mathbin{\mathsf{U}} \alpha)$
6. $\mathsf{F}\, \alpha \equiv \alpha \vee \mathsf{X}(\top \mathbin{\mathsf{U}} \alpha)$
7. $\alpha \mathbin{\mathsf{U}} \beta \equiv \beta \vee (\alpha \wedge \mathsf{X}(\alpha \mathbin{\mathsf{U}} \beta))$
8. $\neg(\alpha \mathbin{\mathsf{U}} \beta) \equiv \neg\beta \wedge (\neg\alpha \vee \neg \mathsf{X}(\alpha \mathbin{\mathsf{U}} \beta))$

Lemma 10.5

Für jede LTL-Formel α gibt es äquivalente Formeln,

die nur aus Atomen, $\neg$, $\wedge$, X und U bestehen,

und es gibt zu α äquivalente Formeln,

die nur aus Atomen, $\perp$, $\rightarrow$, X und U bestehen.

Wir benutzen $\top$, $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, F, G und R weiterhin als abkürzende Schreibweisen.

Definition 10.6 (Erfüllbarkeit, Gültigkeit)

Sei φ eine LTL-Formel.

φ heißt erfüllbar (bezgl. $\models_p$),
falls es eine Belegungsfolge π mit $\pi \models_p \varphi$ gibt.

φ heißt gültig (bezgl. $\models_p$),
falls $\pi \models_p \varphi$ für alle Belegungsfolgen π gilt (Schreibweise: $\models_p \varphi$).

Kripke-Semantik für LTL-Formeln

Definition 10.7 (Erfüllungsrelation $\models_K$ für LTL-Formeln)

Ein Kripke-Modell $\mathcal{M} = (W, R, \xi)$ heißt total,
wenn es für jedes $u \in W$ ein $v \in W$ mit $(u, v) \in R$ gibt,

Sei $\mathcal{M} = (W, R, \xi)$ ein totales Kripke-Modell, $w_0 \in W$ und φ eine LTL-Formel.

$\mathcal{M}, w \models_K \varphi$, falls für jeden unendlichen Pfad $\pi = w_0, w_1, w_2, \ldots$ durch (W, R), der in w_0 beginnt, $\xi(w_0)\,\xi(w_1)\,\xi(w_2)\ldots \models_p \varphi$ gilt.

$\mathcal{M} \models_K \varphi$, falls $\mathcal{M}, w \models_K \varphi$ für jedes $w \in W$.

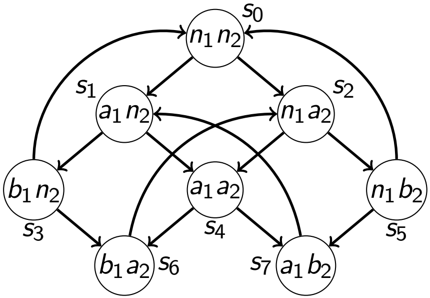
$\models_K \varphi$, falls $\mathcal{M} \models_K \varphi$ für jedes totale Kripke-Modell $\mathcal{M}$.

Die unter Pfad-Semantik gültigen LTL-Formeln sind genau die unter Kripke-Semantik gültigen LTL-Formeln.

Zurück zum Beispiel

Modell $\mathcal{M}_0$ für Behandlung von Ressourcen-Anfragen:

(wir schreiben Atome hier mit kleinen Buchst.)



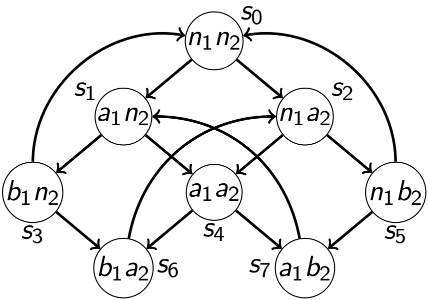
Teste, ob dieses Modell garantiert, dass
jeder Prozess die Ressource – nachdem er sie angefordert hat – auch irgendwann benutzen kann (*fairness*)!

D.h.: gilt $\mathcal{M}_0 \models_{K} \mathsf{G}\,(a_1 \to \mathsf{F}\,b_1) \wedge \mathsf{G}\,(a_2 \to \mathsf{F}\,b_2)$??

Zurück zum Beispiel

Modell $\mathcal{M}_0$ für Behandlung von Ressourcen-Anfragen:

(wir schreiben Atome hier mit kleinen Buchst.)

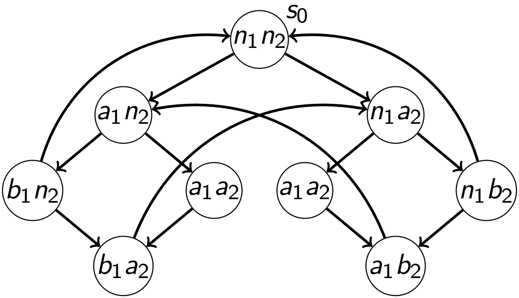


Teste, ob dieses Modell garantiert, dass
jeder Prozess die Ressource – nachdem er sie angefordert hat – auch irgendwann benutzen kann (*fairness*)!

D.h.: gilt $\mathcal{M}_0 \models_K \mathsf{G}\,(a_1 \to \mathsf{F}\,b_1) \wedge \mathsf{G}\,(a_2 \to \mathsf{F}\,b_2)$??

Für Pfad $s_1, s_4, s_7, s_1, s_4, s_7, \ldots$
gilt $\xi(s_1)\xi(s_4)\xi(s_7)\cdots \models_p \mathsf{G}\,a_1$ und $\xi(s_1)\xi(s_4)\xi(s_7)\cdots \not\models_p \mathsf{F}\,b_1$.
Also folgt $\mathcal{M}_0 \not\models_K \mathsf{G}\,(a_1 \to \mathsf{F}\,b_1) \wedge \mathsf{G}\,(a_2 \to \mathsf{F}\,b_2)$.
D.h. die Modellierung durch $\mathcal{M}_0$ ist nicht *fair*.

Die folgende Modellierung $\mathcal{M}_1$ erfüllt die Fairness-Eigenschaft $\mathsf{G}\,(a_1 \to \mathsf{F}\, b_1) \wedge \mathsf{G}\,(a_2 \to \mathsf{F}\, b_2)$.



10.2 Ein Tableau-Kalkül für LTL

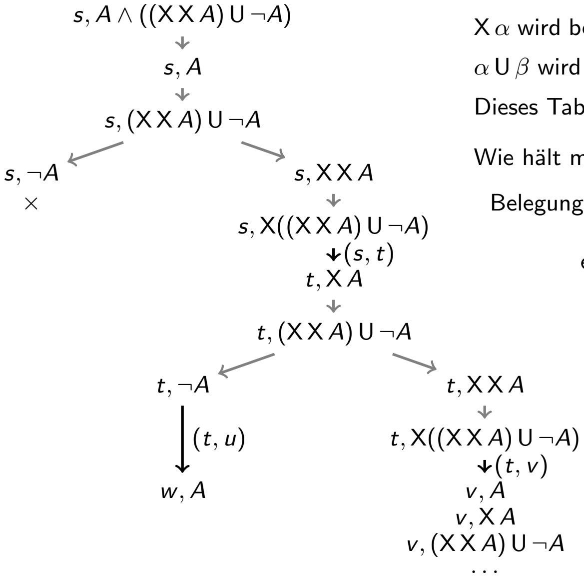
Die Konstruktion eines Tableaus ...

...entspricht in der Aussagenlogik einer systematischen Suche
nach einer Belegung, die die Startformel erfüllt,

...entspricht in der Modallogik einer systematischen Suche
nach einem baumförmigen (endlichen) Kripke-Modell
(also einem Baum aus Belegungen),
das die Startformel in der Startwelt erfüllt.

Entsprechend soll für LTL-Formeln systematisch nach einer unendlichen Folge von Belegungen gesucht werden, die die Startformel erfüllt.

Grobe Idee für ein Tableau ...



$\mathsf{X}\,\alpha$ wird behandelt wie $\Diamond\alpha$ und $\Box\alpha$ (fast ...).

$\alpha\,\mathsf{U}\,\beta$ wird expandiert zu $\beta \vee (\alpha \wedge \mathsf{X}(\alpha\,\mathsf{U}\,\beta))$.

Dieses Tableau wird ziemlich groß ...

Wie hält man es endlich?

Belegungsfolge $\underset{s}{\{A\}}\ \underset{t}{\emptyset}\ \underset{w}{\{A\}}\ \ldots$
erfüllt $A \wedge ((\mathsf{X}\,\mathsf{X}\,A)\,\mathsf{U}\,\neg A)$.

Definition 10.8 (Tableau für LTL-Formeln aus Atomen, $\neg$, $\wedge$, X und U)

Sei φ eine LTL-Formel aus Atomen, $\neg$, $\wedge$, X und U.

1. Ein Knoten, der mit s, φ markiert ist, ist ein Tableau für φ.
2. Sei T ein Tableau für φ, und κ sei ein Knoten in T mit Markierung w, ψ. Wenn ψ eine Formel der Form $\alpha \wedge \beta$, $\neg(\alpha \wedge \beta)$ oder $\neg\neg\alpha$ ist, dann kann κ mit der entsprechenden (lokalen) Expansionsregel expandiert werden.

$w, \alpha \wedge \beta$:
$$\begin{tikzcd} \bullet \arrow[d] \\ w, \alpha \arrow[d] \\ w, \beta \end{tikzcd}$$

$w, \neg(\alpha \wedge \beta)$:
$$\begin{tikzcd} & \bullet \arrow[dl] \arrow[dr] & \\ w, \neg\alpha & & w, \neg\beta \end{tikzcd}$$

$w, \neg\neg\alpha$:
$$\begin{tikzcd} \bullet \arrow[d] \\ w, \alpha \end{tikzcd}$$

Wenn ψ eine Formel der Form $\alpha \,\mathsf{U}\, \beta$ oder $\neg(\alpha \,\mathsf{U}\, \beta)$ ist,
dann kann κ mit folgenden (lokalen) Expansionsregeln expandiert werden.

$w, \alpha \,\mathsf{U}\, \beta$:
$$\begin{tikzcd} & \bullet \arrow[dl] \arrow[dr] & \\ w, \beta & & w, \alpha \arrow[d] \\ & & w, \mathsf{X}(\alpha \,\mathsf{U}\, \beta) \end{tikzcd}$$

$w, \neg(\alpha \,\mathsf{U}\, \beta)$:
$$\begin{tikzcd} & \bullet \arrow[d] & \\ & w, \neg\beta \arrow[dl] \arrow[dr] & \\ w, \neg\alpha & & w, \neg\mathsf{X}(\alpha \,\mathsf{U}\, \beta) \end{tikzcd}$$

3. Sei T ein Tableau für φ, und π sei ein Pfad durch T, der mit Markierungen für eine Welt w endet. Alle Knoten mit Markierungen w, α, die nach einer Regel unter 2.) expandiert werden können, seien bereits expandiert (d.h. alle lokalen Expansionen für die Welt w wurden durchgeführt).
Seien $w, \mathsf{X}\,\alpha_1, \ldots, w, \mathsf{X}\,\alpha_q,\ w, \neg\,\mathsf{X}\,\beta_1, \ldots, w, \neg\,\mathsf{X}\,\beta_r$ alle nicht expandierten Knoten für die Welt w, deren Formeln mit X oder $\neg\,\mathsf{X}$ beginnen.
Dann werden alle diese Knoten mit folgender Regel expandiert.
Dabei ist z eine neue Welt, die bisher nicht im Tableau vorkommt.

$$\begin{array}{c} \bullet \\ \big\downarrow (w,z) \\ z, \alpha_1 \\ \downarrow \\ \vdots \\ z, \alpha_q \\ \downarrow \\ z, \neg\beta_1 \\ \downarrow \\ \vdots \\ z, \neg\beta_r \end{array}$$

Durch Expansion entsteht ein (weiteres) Tableau für φ.

Wie wird dieses Tableau widersprüchlich?

$$s, \neg A \wedge ((\mathsf{X}\,\neg A)\,\mathsf{U}\,A)$$

$$s, \neg A$$

$$s, (\mathsf{X}\,\neg A)\,\mathsf{U}\,A$$

$$s, A \qquad\qquad s, \mathsf{X}\,\neg A$$

$$\times$$

$$s, \mathsf{X}((\mathsf{X}\,\neg A)\,\mathsf{U}\,A)$$

$$\downarrow (s, t)$$

$$t, \neg A$$

$$t, (\mathsf{X}\,\neg A)\,\mathsf{U}\,A$$

$$t, A \qquad\qquad t, \mathsf{X}\,\neg A$$

$$\times$$

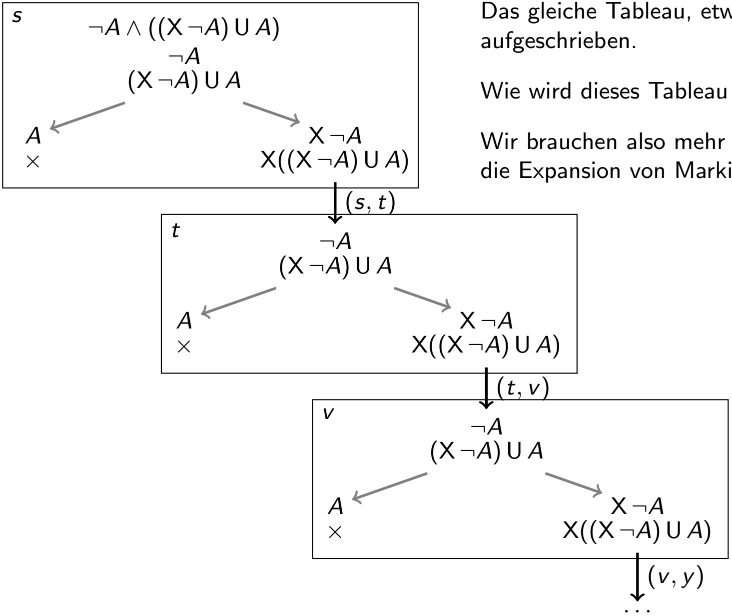
$$t, \mathsf{X}((\mathsf{X}\,\neg A)\,\mathsf{U}\,A)$$

$$\downarrow (t, v)$$

$$v, \neg A$$

$$v, (\mathsf{X}\,\neg A)\,\mathsf{U}\,A$$

$$\ldots$$



Das gleiche Tableau, etwas lesbarer aufgeschrieben.

Wie wird dieses Tableau geschlossen?

Wir brauchen also mehr Eigenschaften, um die Expansion von Markierungen zu beenden.

Wir müssen jetzt Bedingungen zum Beenden der Expansion eines Pfades formulieren.
Dafür brauchen wie ein paar Begriffe.

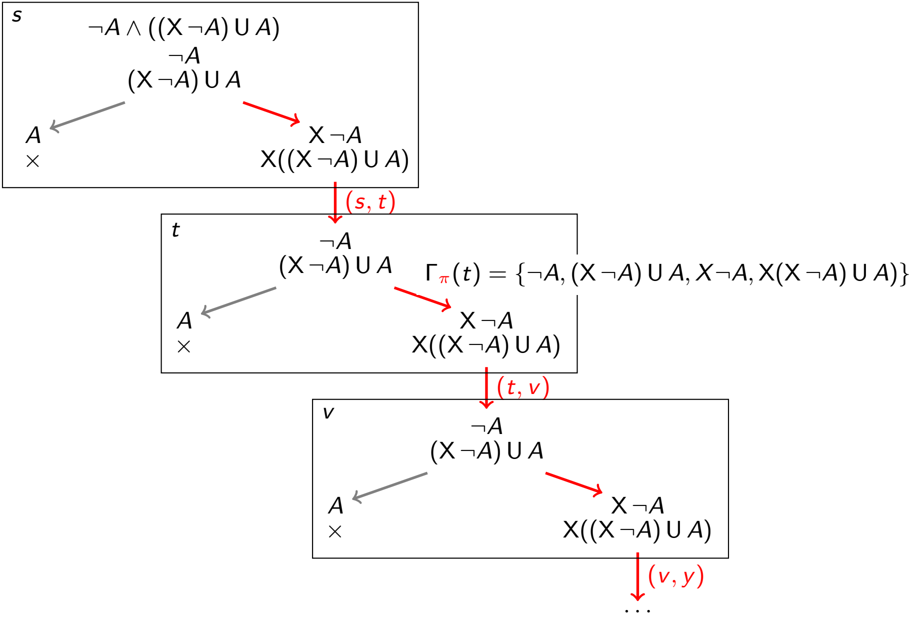
Sei π ein Pfad durch ein endliches Tableau.

$W_\pi = \{u \mid \text{eine Knotenmarkierung } u, \alpha \text{ kommt auf } \pi \text{ vor}\}$ ist die Menge aller Welten von π.

$R_\pi = \{(u, v) \mid \text{Kantenmarkierung } (u, v) \text{ kommt auf } \pi \text{ vor}\}$ ist die Menge aller Kanten von π.

(W_π, R_π) ist ein (endlicher) Pfad.

Für $u \in W_\pi$ ist $\Gamma_\pi(u)$ die Menge aller Formeln in Markierungen mit u,
$\Gamma_\pi(u) = \{\alpha \mid \text{eine Knotenmarkierung } u, \alpha \text{ kommt auf } \pi \text{ vor}\}$.



s
¬A ∧ ((X ¬A) U A)
¬A
(X ¬A) U A
A
×
X ¬A
X((X ¬A) U A)
(s, t)
t
¬A
(X ¬A) U A
Γπ(t) = {¬A, (X ¬A) U A, X¬A, X(X ¬A) U A)}
A
×
X ¬A
X((X ¬A) U A)
(t, v)
v
¬A
(X ¬A) U A
A
×
X ¬A
X((X ¬A) U A)
(v, y)
...

Die Expansion eines Pfades π kann beendet werden,

(1) wenn in einer Welt widersprüchliche Formeln in den Markierungen stehen (das ist wie bisher: widersprüchlicher Pfad), oder

(2) wenn man sich in einer Schleife von Belegungen befindet, in der zu erfüllende until-Formeln nie erfüllt werden.

$$
\begin{aligned}
P_\pi = \big\{ w \mid\ & \text{es gibt } u, v \in W_\pi \text{ mit} \\
& (u,v) \in R_\pi^+ \text{ und } (v,w) \in R_\pi^+ \text{ und } \Gamma_\pi(u) = \Gamma_\pi(v) = \Gamma_\pi(w), \text{ und} \\
& (1) \text{ es gibt ein } \mathsf{X}(\alpha \,\mathsf{U}\, \beta) \in \Gamma_\pi(u), \text{ so dass für alle } x \in W_\pi \text{ mit} \\
& \qquad (v,x) \in R_\pi^+,\ (x,w) \in R_\pi^* \text{ gilt: } \beta \notin \Gamma_\pi(x), \text{ und} \\
& (2) \text{ für jedes } \mathsf{X}(\alpha \,\mathsf{U}\, \beta) \in \Gamma_\pi(u), \text{ für das es ein } x \in W_\pi \text{ gibt mit} \\
& \qquad (v,x) \in R_\pi^+,\ (x,w) \in R_\pi^* \text{ und } \beta \in \Gamma_\pi(x), \\
& \text{gibt es ein } y \in W_\pi \text{ mit } (u,y) \in R_\pi^+,\ (y,v) \in R_\pi^* \text{ und } \beta \in \Gamma_\pi(y) \big\}
\end{aligned}
$$

R_π^* ist die reflexive und transitive Hülle von R_π,
R_π^+ ist die transitive Hülle von R_π.

Definition 10.9 (erfolgloser Pfad eines LTL-Tableaus)

Ein Pfad π durch ein LTL-Tableau heißt erfolglos,
wenn er widersprüchlich ist oder wenn $P_\pi \neq \emptyset$.

Die Expansion eines Pfades kann auch beendet werden,
wenn man sich in einer Schleife von Belegungen befindet,
in der alle zu erfüllenden until-Formeln erfüllt sind.

$$L_\pi = \{(v,u) \mid (u,v) \in R_\pi^+, \Gamma_\pi(v) \subseteq \Gamma_\pi(u) \text{ und für alle } \mathsf{X}(\alpha \,\mathsf{U}\, \beta) \in \Gamma_\pi(u) \text{ gibt es einen Knoten } w \text{ mit } (u,w) \in R_\pi^+ \text{ und } (w,v) \in R_\pi^* \text{ und } \beta \in \Gamma_\pi(w)\}$$

Definition 10.10 (erfolgreicher Pfad eines LTL-Tableaus)

Ein Pfad π durch ein LTL-Tableau heißt erfolgreich,
(1) wenn er nicht erfolglos ist und alle Knoten des Pfades expandiert sind, oder
(2) wenn $L_\pi \neq \emptyset$.

Definition 10.11 (Tableau-beweisbare LTL-Formeln $\vdash_{\text{t-Tab}}$)

Sei α eine LTL-Formel. α heißt Tableau-beweisbar (Schreibweise $\vdash_{\text{t-Tab}} \alpha$), falls $\neg\alpha$ ein LTL-Tableau besitzt, das nur aus erfolglosen Pfaden besteht.

Definition 10.12 (systematisches LTL-Tableau)

Ein LTL-Tableau heißt systematisch, falls alle Knoten auf allen Pfaden expandiert werden, bis der Pfad erfolgreich oder erfolglos ist.

Lemma 10.13 (endliche LTL-Tableaux reichen)

Jede LTL-Formel α besitzt ein endliches systematisches Tableau.

Satz 10.14 (Vollständigkeitssatz für $\vdash_{\text{t-Tab}}$)

Sei α eine LTL-Formel.
Dann gilt $\models_{p} \alpha$ genau dann, wenn $\vdash_{\text{t-Tab}} \alpha$.

[Literatur:
Mark Reynolds:
A traditional tree-style tableau for LTL
http://www.csse.uwa.edu.au/∼mark/research/Online/ltlsattabLONG.pdf, 2015

Agnes Schubert:
Vollständigkeit und Korrektheit für ein Tableau-Verfahren für LTL. BSc-Arbeit, 2016]

10.3 Ein Frege-Kalkül für LTL

Wir wollen nur einen Eindruck der nötigen Axiome vermitteln und erlauben uns daher großzügig abkürzende Schreibweisen ...

Definition 10.15 (LTL-Frege-Kalkül)

1. Die *Elemente* des LTL-Frege-Kalküls sind LTL-Formeln.
2. Die *Axiome* sind für alle Formeln α, β, φ:

   (A1) $\alpha \to (\beta \to \alpha)$

   (A2) $(\alpha \to (\beta \to \varphi)) \to ((\alpha \to \beta) \to (\alpha \to \varphi))$

   (A3) $\neg\neg\alpha \to \alpha$

   (U) $\alpha \,\mathsf{U}\, \beta \leftrightarrow \beta \vee (\alpha \wedge \mathsf{X}(\alpha \,\mathsf{U}\, \beta))$

   (X) $\neg \mathsf{X}\, \alpha \leftrightarrow \mathsf{X} \neg\alpha$
3. Die *Schlussregeln* sind:

$$\frac{\alpha \qquad \alpha \to \beta}{\beta} \qquad \frac{\alpha}{\mathsf{X}\,\alpha} \qquad \frac{\varphi \to (\beta \wedge (\alpha \vee \mathsf{X} \neg(\neg(\alpha \vee \varphi) \,\mathsf{U}\, \neg(\beta \vee \varphi))))}{\varphi \to \neg(\neg\alpha \,\mathsf{U}\, \neg\beta)}$$

4. Herleitungen sind definiert wie üblich.

[Literatur:
Martin Lange:
A quick axiomatization of LTL with past
Mathematical Logic Quarterly 51, No. 1, 83–88 (2005)]

Was haben wir in Vorlesung 10 gelernt?

- Wir kennen die temporalen Operatoren F, X, G, U und R der linearen Zeitlogik LTL.
- Wir kennen die Pfad-Semantik und die Kripke-Semantik von LTL-Formeln.
- Wir können Tableaux für LTL-Formeln konstruieren und kennen die Relation $\vdash_{\text{t-Tab}}$ der Tableau-Beweisbarkeit.
- Wir haben eine Vorstellung davon, wie man in endlichen Tableaux unendliche erfüllende Pfade für LTL-Formeln finden kann.
- Wir haben einen Frege-Kalkül für LTL gesehen.

Vorlesung 11: Büchi-Automaten

[Literatur: Hoffmann, Lange: Automatentheorie und Logik.]

Beim Tableauverfahren sucht man nach erfüllenden *unendlichen* Belegungsfolgen für LTL-Formeln. Bisher sind wir den Umgang mit solchen unendlichen Objekten nicht gewohnt. Wir werden sehen, dass man alles für uns Nötige mit Büchi-Automaten – das sind endliche Automaten für unendliche Wörter – machen kann. Das ist nur ein kleiner Schritt weg von dem, was wir bereits kennen.

Uns interessieren hier insbesondere verallgemeinerte Büchi-Automaten (die nicht mehr können als Büchi-Automaten, aber für unsere Zwecke besser zu benutzen sind) und ein Algorithmus, der das Leerheitsproblem für Büchi-Automaten entscheidet.

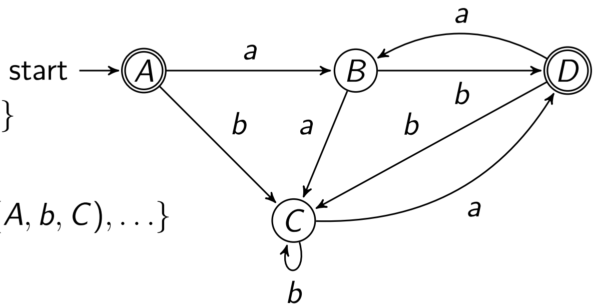
In Vorlesung 12 werden wir dann einen engen Zusammenhang zwischen LTL-Formeln und (verallgemeinerten) Büchi-Automaten sehen: man kann aus LTL-Formeln Büchi-Automaten basteln (ähnlich wie die Tableau-Konstruktion), die genau die erfüllenden Belegungsfolgen der Formeln akzeptieren. Der Büchi-Automat für eine Formel $\neg\varphi$ akzeptiert also nichts, wenn φ gültig ist. Somit erhalten wir durch den Konstruktions-Algorithmus für den Büchi-Automaten und den Algorithmus für das Leerheitsproblem einen Algorithmus zum Gültigkeitstest für LTL-Formeln.

Die Vorlesung 11 ist also eine Vorbereitung für Vorlesung 12.

11.1 Endliche Automaten und reguläre Sprachen

Ein *endlicher Automat* $M = (Q, \Sigma, \Delta, q_0, F)$ besteht aus

- einer endlichen Menge Q (Zustände)
- einem endlichen Alphabet Σ
- einer Relation $\Delta \subseteq Q \times \Sigma \times Q$ (Zustandsübergänge)
- einem Startzustand $q_0 \in Q$
- einer Menge von Endzuständen $F \subseteq Q$



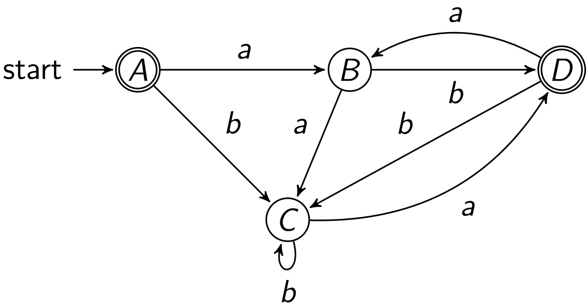
$$\begin{aligned}
Q &= \{A, B, C, D\} \\
\Sigma &= \{a, b\} \\
\Delta &= \{(A, a, B), (A, b, C), \ldots\} \\
q_0 &= A \\
F &= \{A, D\}
\end{aligned}$$

Von einem EA akzeptierte Wörter

Ein Wort $w = a_1 \cdots a_n$ ($a_i \in \Sigma$) wird vom endlichen Automaten M *akzeptiert*,

wenn es eine Zustandsfolge $\rho_0\rho_1\rho_2 \cdots \rho_n$ gibt ($\rho_i \in Q$), so dass

- $\rho_0 = q_0$ (die Folge beginnt mit dem Startzustand),
- $(\rho_{i-1}, a_i, \rho_i) \in \Delta$ für $1 \leqslant i \leqslant n$
  (die Zustandsübergänge entsprechen dem Wort w), und
- $\rho_n \in F$ (die Folge endet in einem Endzustand).



Determinismus vs. Nichtdeterminismus

Ein EA $M = (Q, \Sigma, \Delta, q_0, F)$ akzeptiert die Sprache

$$L(M) = \{w \in \Sigma^* \mid M \text{ akzeptiert } w\} \ .$$

Ein EA heißt *deterministisch*, wenn Δ eine Funktion $Q \times \Sigma \to Q$ ist.

Satz 11.1 (deterministische und nichtdeterministische EAen können das gleiche)

Sei N ein EA.
Dann gibt es einen deterministischen EA M mit $L(N) = L(M)$.

Reguläre Sprachen und ihre Abschlusseigenschaften

Eine Sprache heißt *regulär*, wenn sie von einem EA akzeptiert wird.

Satz 11.2

Die Klasse der regulären Sprachen ist abgeschlossen unter Komplement, Vereinigung, Konkatenation und Sternbildung.

Entscheidungsprobleme über EAen

Definition 11.3 (Leerheitsproblem für EAen E_{NFA})

Das Leerheitsproblem für endliche Automaten E_{NFA} ist das folgende Entscheidungsproblem.

gegeben:	ein endlicher Automat M
gefragt:	ist $L(M) = \emptyset$?

Definition 11.4 (Nicht-Leerheitsproblem für EAen $\overline{E_{NFA}}$)

Das Nicht-Leerheitsproblem für endliche Automaten ist das folgende Entscheidungsproblem.

gegeben:	ein endlicher Automat M
gefragt:	ist $L(M) \neq \emptyset$?

Algorithmus für das Nicht-Leerheitsproblem

für endliche Automaten

Eingabe: B (* B ist EA, $B = (Q, \Sigma, \Delta, q_0, F)$ *)

$s := q_0$

wiederhole $|Q| - 1$-mal { (* finde einen Weg von q_0 mit Länge $\leqslant |Q| - 1$ *)

  wähle nichtdeterministisch $v \in Q$

  falls $(s, a, v) \in \Delta$ für ein $a \in \Sigma$

    dann $s := v$

}

falls $s \in F$

  dann akzeptiere

  sonst verwirf

Algorithmus für das Nicht-Leerheitsproblem

für endliche Automaten

Eingabe: B (* B ist EA, $B = (Q, \Sigma, \Delta, q_0, F)$ *)

$s := q_0$

wiederhole $|Q| - 1$-mal { (* finde einen Weg von q_0 mit Länge $\leqslant |Q| - 1$ *)

  wähle nichtdeterministisch $v \in Q$

  falls $(s, a, v) \in \Delta$ für ein $a \in \Sigma$

    dann $s := v$

}

falls $s \in F$

  dann akzeptiere

  sonst verwirf

Wenn $L(B) \neq \emptyset$ (d.h. B akzeptiert irgendein Wort),
dann gibt es einen Weg vom Startzustand q_0 zu einem Endzustand in F.
Es gibt dann auch einen solchen Weg mit Länge $\leqslant |Q| - 1$
– sonst kommt ein Zustand doppelt vor und man kann die Schleife rausschneiden.
Der Berechnungspfad des Alg., der diesem Weg entspricht, akzeptiert.

Algorithmus für das Nicht-Leerheitsproblem

für endliche Automaten

Eingabe: B (* B ist EA, $B = (Q, \Sigma, \Delta, q_0, F)$ *)

$s := q_0$

wiederhole $|Q| - 1$-mal { (* finde einen Weg von q_0 mit Länge $\leqslant |Q| - 1$ *)

  wähle nichtdeterministisch $v \in Q$

  falls $(s, a, v) \in \Delta$ für ein $a \in \Sigma$

    dann $s := v$

}

falls $s \in F$

  dann akzeptiere

  sonst verwirf

Wenn $L(B) = \emptyset$ (d.h. B akzeptiert kein Wort),
dann gibt es keinen Weg vom Startzustand q_0 zu einem Endzustand in F.
Also verwirft jeder Berechnungspfad des Algorithmus.

Damit ist gezeigt, dass der Algorithmus $\overline{E_{NFA}}$ entscheidet.

Algorithmus für das Nicht-Leerheitsproblem

für endliche Automaten

Eingabe: B (* B ist EA, $B = (Q, \Sigma, \Delta, q_0, F)$ *)

$s := q_0$

wiederhole $|Q| - 1$-mal $\{$ (* finde einen Weg von q_0 mit Länge $\leqslant |Q| - 1$ *)

  wähle nichtdeterministisch $v \in Q$

  falls $(s, a, v) \in \Delta$ für ein $a \in \Sigma$

    dann $s := v$

$\}$

falls $s \in F$

  dann akzeptiere

  sonst verwirf

Der benötigte Speicherplatz des Algorithmus ist der Platz für die Variablen s und v für Zustände von B und für einen Schleifenzähler für Werte $0, 1, 2, \ldots, |Q|$.
Für jede Variable braucht man höchstens (etwa) $\log |Q|$ Bit.
Für die Darstellung von B in der Eingabe braucht man mindestens $|Q|$ Bit.
Also ist der Speicherbedarf $O(\log n)$ bei Eingabelänge n.

Komplexität von E_{NFA} und $\overline{E_{NFA}}$ (ein kleiner Exkurs)

Insgesamt folgt, dass $\overline{E_{NFA}}$ in der Komplexitätsklasse NL ist.

> *NL ist die Klasse aller Entscheidungsprobleme, die von nichtdeterministischen Turingmaschinen mit Platz $O(\log n)$ berechnet werden können.*
> *Die Turingmaschinen haben ein Eingabeband, auf dem nicht geschrieben werden darf, und Arbeitsbänder zum Lesen und Schreiben.*

Man weiß auch, dass $\overline{E_{NFA}}$ nicht besser zu lösen ist, außer wenn Komplexitätsklassen zusammenfallen (ähnlich P=NP).
Das drückt man mit dem Begriff der *NL-Vollständigkeit* aus.

Außerdem weiß man, dass NL unter Komplement abgeschlossen ist.

Satz 11.5

E_{NFA} und $\overline{E_{NFA}}$ sind NL-*vollständig.*

Komplexität von E_{NFA} und $\overline{E_{NFA}}$ (ein kleiner Exkurs)

Insgesamt folgt, dass $\overline{E_{NFA}}$ in der Komplexitätsklasse NL ist.
Man weiß auch, dass $\overline{E_{NFA}}$ nicht besser zu lösen ist, außer wenn Komplexitätsklassen zusammenfallen (ähnlich P=NP).
Das drückt man mit dem Begriff der *NL-Vollständigkeit* aus.

Außerdem weiß man, dass NL unter Komplement abgeschlossen ist.

Satz 11.5

E_{NFA} und $\overline{E_{NFA}}$ sind NL-*vollständig.*

$$\begin{tikzcd}
 & & & & \mathrm{NP} \arrow[dr] & \\
\mathrm{NC}^1 \arrow[r] & \mathrm{L} \arrow[r] & \mathrm{NL} \arrow[r] & \mathrm{P} \arrow[ur] \arrow[dr] & & \mathrm{PSPACE} \\
 & & & & \mathrm{coNP} \arrow[ur] &
\end{tikzcd}$$

polynomielle Zeit

logarithmischer Platz polynomieller Platz

11.2 Büchi-Automaten und ω-reguläre Sprachen

Endliche Automaten für unendliche Wörter

Sei Σ ein Alphabet (also eine endliche Menge und nicht leer).

Σ^* ist die Menge aller *endlichen* Wörter über Σ.

Σ^* ist abzählbar unendlich.

Σ^ω ist die Menge aller *unendlichen* Wörter über Σ. („ω-Wörter“)

Σ^ω ist überabzählbar unendlich.

Noch allgemeiner: sei S eine Menge.

$S^* = \{w_1 w_2 \cdots w_m \mid m \geqslant 0 \text{ und } w_i \in S \text{ für alle } i = 1, 2, \ldots, m\}$

$S^\omega = \{w_1 w_2 \cdots \mid w_i \in S \text{ für alle } i \geqslant 1\}$

Büchi-Automaten

Ein *Büchi-Automat* ist ein endlicher Automat $M = (Q, \Sigma, \Delta, q_0, F)$.

Ein ω-Wort $w = a_1 a_2 \cdots$ $(a_i \in \Sigma)$ wird vom Büchi-Automaten M *akzeptiert*,
wenn es eine unendliche Zustandsfolge $\rho_0 \rho_1 \rho_2 \cdots$ gibt $(\rho_i \in Q)$, so dass

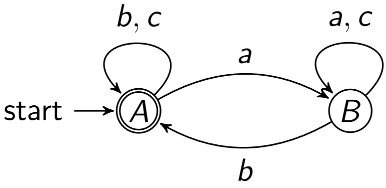
- $\rho_0 = q_0$ (die Folge beginnt mit dem Startzustand),
- $(\rho_{i-1}, a_i, \rho_i) \in \Delta$ für alle $i \geqslant 1$
  (die Zustandsübergänge entsprechen dem Wort w), und
- es gibt unendlich viele i mit $\rho_i \in F$
  (die Folge durchläuft unendlich oft einen Endzustand).

$L_\omega(M) = \{w \in \Sigma^\omega \mid M \text{ akzeptiert } w\}$
ist die vom Büchi-Automat M akzeptierte Sprache.

Beispiel 1

$\Sigma = \{a, b, c\}$

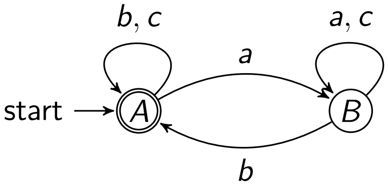
M_1:



Beispiel 1

$\Sigma = \{a, b, c\}$

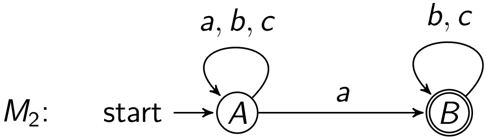
M_1:



$L_\omega(M_1) = \{w \in \Sigma^\omega \mid \text{in } w \text{ kommt nach jedem } a \text{ irgendwann ein } b\}$

Beispiel 2

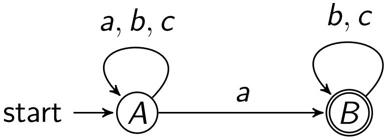
$\Sigma = \{a, b, c\}$



Beispiel 2

$\Sigma = \{a, b, c\}$

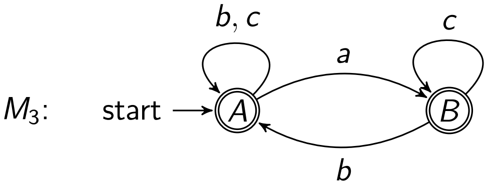
M_2:



$L_\omega(M_2) = \{w \in \Sigma^\omega \mid \text{in } w \text{ gibt es ein letztes } a\}$

Beispiel 3

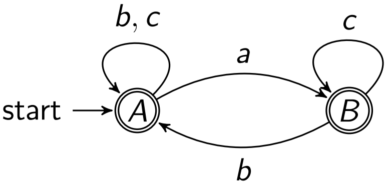
$\Sigma = \{a, b, c\}$



Beispiel 3

$\Sigma = \{a, b, c\}$

M_3:

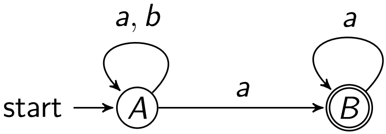


$L_\omega(M_3) = \{w \in \Sigma^\omega \mid \text{zwischen zwei } a \text{ in } w \text{ kommt stets ein } b \text{ vor}\}$

Beispiele 4 und 5

$\Sigma = \{a, b\}$

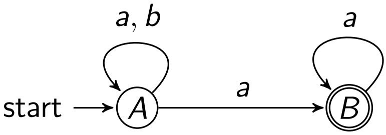
M_4:



Beispiele 4 und 5

$\Sigma = \{a, b\}$

M_4:

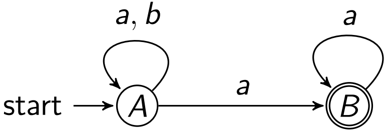


$L_\omega(M_4) = \{w \in \Sigma^\omega \mid \text{in } w \text{ gibt es nur endlich viele } b\}$

Beispiele 4 und 5

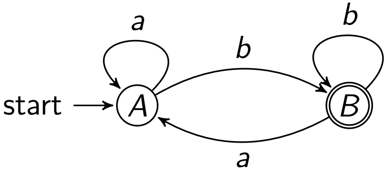
$\Sigma = \{a, b\}$

M_4:



$L_\omega(M_4) = \{w \in \Sigma^\omega \mid \text{in } w \text{ gibt es nur endlich viele } b\}$

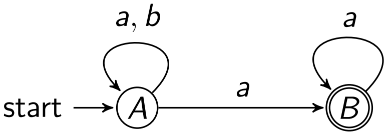
M_5:



Beispiele 4 und 5

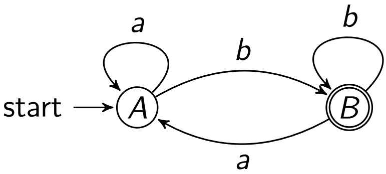
$\Sigma = \{a, b\}$

M_4:



$$L_\omega(M_4) = \{w \in \Sigma^\omega \mid \text{in } w \text{ gibt es nur endlich viele } b\}$$

M_5:

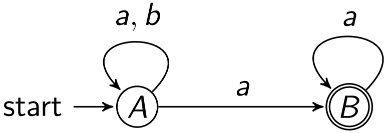


$$L_\omega(M_5) = \{w \in \Sigma^\omega \mid \text{in } w \text{ kommen unendlich viele } b \text{ vor}\}$$

Beispiele 4 und 5

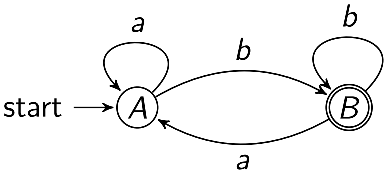
$\Sigma = \{a, b\}$

M_4:



$$L_\omega(M_4) = \{w \in \Sigma^\omega \mid \text{in } w \text{ gibt es nur endlich viele } b\}$$

M_5:



$$L_\omega(M_5) = \{w \in \Sigma^\omega \mid \text{in } w \text{ kommen unendlich viele } b \text{ vor}\} \;\; = \overline{L_\omega(M_4)}$$

Determinismus vs. Nichtdeterminismus

Lemma 11.6 (nicht-deterministische Büchi-Automaten können mehr als deterministische Büchi-Automaten)

Es gibt keinen deterministischen Büchi-Automaten für

$$L_\omega(M_4) = \{w \in \{a,b\}^\omega \mid \textit{in } w \textit{ gibt es nur endlich viele } b\}.$$

Beweis: Sei B ein det. Büchi-Automat mit z Zuständen und $L_\omega(B) \supseteq L_\omega(M_4)$.
Da $baaa\cdots \in L_\omega(M_4)$, gibt es ein r_1,
so dass B bei Eingabe ba^{r_1} in einem Endzustand ist.
Entsprechend gibt es ein Wort $ba^{r_1}ba^{r_2}ba^{r_3}\ldots ba^{r_{z+1}}$, so dass B bei Eingabe von $ba^{r_1}\ldots ba^{r_i}$ in einem Endzustand ist (für jedes $i = 1, 2, \ldots, z+1$).
Da B nur z Zustände hat, muss einer dieser Zustände
bei Eingabe von $ba^{r_1}\ldots ba^{r_x}$ und auch bei Eingabe von $ba^{r_1}\ldots ba^{r_y}$ $(x < y)$ erreicht sein.
Dann ist $ba^{r_1}\ldots ba^{r_x}(ba^{r_{x+1}}\ldots ba^{r_y})^\omega$ ein ω-Wort mit unendlich vielen b,
das von B akzeptiert wird.
Also gilt $L_\omega(B) \supsetneq L_\omega(M_4)$.

Folglich gibt es keinen deterministischen Büchi-Automaten für $L_\omega(M_4)$. ✓

ω-reguläre Sprachen

Sei Σ ein Alphabet.
Eine Teilmenge $A \subseteq \Sigma^\omega$ heißt *ω-regulär*,
falls es einen Büchi-Automaten B mit $A = L_\omega(B)$ gibt.

(Mit Büchi-Automat ist ein nichtdeterministischer Büchi-Automat gemeint.)

Lemma 11.7 (Abschlusseigenschaften ω-regulärer Sprachen)

1. *Wenn A regulär ist, dann ist A^ω ω-regulär.*
2. *Wenn A regulär und B ω-regulär ist, dann ist $A\,B$ ω-regulär.*
3. *Die Klasse der ω-regulären Sprachen ist abgeschlossen unter Durchschnitt und Komplement.*

Abschlusseig. ω-regulärer Sprachen

Der Beweis der Komplementabgeschlossenheit der Klasse der ω-regulärer Sprachen ist schwierig.
Wichtig ist, dass der Komplement-Automat exponentiell wachsen kann.

Zum Abschluss unter Durchschnitt:
Bezeichne $w[i]$ den Präfix der Länge i von w.
Seien A und B zwei Büchi-Automaten.
Das ω-Wort w wird von A und B akzeptiert genau dann, wenn es eine unendliche Folge $i_1 < i_2 < i_3 < \dots$ von natürlichen Zahlen gibt, so dass

- für jedes ungerade ℓ gilt: A erreicht mit $w[i_\ell]$ einen Endzustand, und
- für jedes gerade ℓ gilt: B erreicht mit $w[i_\ell]$ einen Endzustand.

Aus $A = (Q_A, \Sigma, \Delta_A, z_A, F_A)$ und $B = (Q_B, \Sigma, \Delta_B, z_B, F_B)$ ergibt sich damit
$C = (Q_A \times Q_B \times \{1,2\}, \Sigma, \Delta_C, (z_A, z_B, 1), F_A \times Q_B \times \{1\} \cup Q_A \times F_B \times \{2\})$
mit $\Delta_C = \{((a,b,i), x, (a',b',j)) \mid (a,x,a') \in \Delta_A$ und $(b,x,b') \in \Delta_B$ und
(1) $i = 1$ und $a \in F_A$ und $j = 2$, oder
(2) $i = 2$ und $b \in F_B$ und $j = 1$, oder (3) $i = j$ $\}$

Entscheidungsprobleme über Büchi-Automaten

Definition 11.8 (Leerheitsproblem für Büchi-Automaten E_{BA})

Das Leerheitsproblem für Büchi-Automaten ist folgendes Entscheidungsproblem.

gegeben: ein Büchi-Automat B
gefragt: ist $L_\omega(B) = \emptyset$?

Definition 11.9 (Nicht-Leerheitsproblem für Büchi-Autom. $\overline{E_{BA}}$)

Das Nicht-Leerheitsproblem für Büchi-Automaten ist folgendes Entscheidungsproblem.

gegeben: ein Büchi-Automat B
gefragt: ist $L_\omega(B) \neq \emptyset$?

Algorithmus für das Nicht-Leerheitsproblem

für Büchi-Automaten

Idee:

wenn Büchi-Automat B ein Wort w akzeptiert,
dann gibt es *einen* Endzustand, der bei w unendlich oft durchlaufen wird.

Algorithmus dazu:
(1) wähle nichtdet. einen Endzustand q
(2) zeige, dass q vom Startzustand aus erreichbar ist
(3) zeige, dass q auf einem Kreis liegt

Wenn der Algorithmus das gezeigt hat, dann ist $L_\omega(B)$ nicht leer
(und umgekehrt).

Algorithmus für das Nicht-Leerheitsproblem

für Büchi-Automaten

Eingabe Büchi-Automat $B = (Q, \Sigma, \Delta, q_0, F)$
wähle nichtdeterministisch $q \in F$
$s := q_0$
wiederhole $|Q| - 1$-mal { (∗ Finde einen Weg von q_0 nach q. ∗)
  wähle nichtdeterministisch $v \in Q$
  falls $(s, a, v) \in \Delta$ für ein $a \in \Sigma$, dann $s := v$
}
falls $s \neq q$: dann verwirf (und halte)
wähle nichtdeterministisch $v \in Q$ (∗ Finde einen Weg von q nach q, ∗)
falls $(s, a, v) \in \Delta$ für ein $a \in \Sigma$, dann $s := v$ (∗ der mindestens eine Kante enthält. ∗)
  sonst verwirf (und halte)
wiederhole $|Q| - 2$-mal
  wähle nichtdeterministisch $v \in Q$
  falls $(s, a, v) \in \Delta$ für ein $a \in \Sigma$, dann $s := v$
}
falls $s = q$
  dann akzeptiere
  sonst verwirf

Analyse des Algorithmus

1. Der Algorithmus hält bei jeder Eingabe und ist korrekt.
2. Der Wert jeder Variable kann mit $\log |Q|$ Bit gespeichert werden.
3. Der Algorithmus ist nichtdeterministisch.

Wir bekommen die gleichen Ergebnisse wie für EAen.

Satz 11.10

E_{BA} und $\overline{E_{BA}}$ sind NL-*vollständig.*

Verallgemeinerte Büchi-Automaten

werden uns das Leben einfacher machen

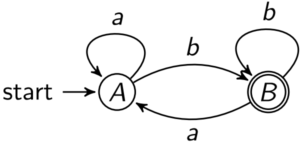
Ein *verallgemeinerter Büchi-Automat* $B = (Q, \Sigma, \Delta, q_0, \mathcal{F})$ besteht aus

- einer endlichen Menge Q (Zustände)
- einem endlichen Alphabet Σ
- einer Relation $\Delta \subseteq Q \times \Sigma \times Q$ (Zustandsübergänge)
- einem Startzustand $q_0 \in Q$
- einer Menge $\mathcal{F} \subseteq \mathcal{P}(Q)$ aus Teilmengen von Q

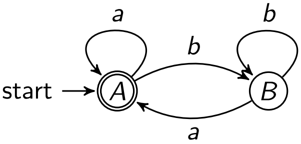
Ein ω-Wort $w \in \Sigma^\omega$ wird von B *akzeptiert*,
falls es eine Zustandsfolge $s_0, s_1, s_2, \ldots$ gibt, die von B bei Eingabe w durchlaufen wird,
in der für jedes $F \in \mathcal{F}$ unendlich viele i mit $s_i \in F$ vorkommen.

Beispiele für verallg. Büchi-Automaten

B_1:

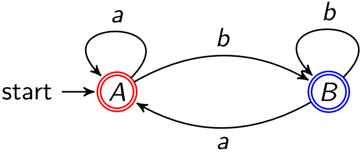


B_2:



$L_\omega(B_1) = \{w \in \{a,b\}^\omega \mid \text{ in } w \text{ kommen unendlich viele } b \text{ vor}\}$
$L_\omega(B_2) = \{w \in \{a,b\}^\omega \mid \text{ in } w \text{ kommen unendlich viele } a \text{ vor}\}$

B_3:



$L_\omega(B_3) = \{w \in \{a,b\}^\omega \mid \text{ in } w \text{ kommen unendl. viele } a \text{ und unendl. viele } b \text{ vor}\}$

B_3 ist ein verallgemeinerter Büchi-Automat mit $\mathcal{F} = \big\{\{A\},\{B\}\big\}$.

Lemma 11.11

Jede ω-Sprache, die von einem verallgemeinerten Büchi-Automaten akzeptiert wird, wird auch von einem (normalen) Büchi-Automaten akzeptiert.

Beweisidee:
Sei $B = (Q, \Sigma, \Delta, q_0, \{F_0, F_1, \ldots, F_{\ell-1}\})$ ein verallgemeinerter Büchi-Automat.
Definiere $\hat{B} = (Q \times \{0, 1, \ldots, \ell - 1\}, \Sigma, \Delta', (q_0, 0), F)$ mit $((b, k), x, (b', k')) \in \Delta'$ genau dann, wenn

- $(b, x, b') \in \Delta$ (Übergang gemäß B) und
- wenn $b \in F_k$, dann $k' = (k + 1) \bmod \ell$; sonst ist $k = k'$.
  (Wenn ein Endzustand in F_k erreicht wurde,
  dann wird auf einen Endzustand in F_{k+1} gewartet.)

Schließlich ist $F = \bigcup_{i=0,1,\ldots,\ell-1} F_i \times \{i\}$.

Es ist nicht zu schwer zu sehen, dass $L_\omega(B) = L_\omega(\hat{B})$. ✓

Was haben wir in Vorlesung 11 gelernt?

- Wir haben unendliche Wörter, Büchi-Automaten und verallgemeinerte Büchi-Automaten kennengelernt.
- Wir kennen die Unterschiede zwischen (klassischen) endlichen Automaten und Büchi-Automaten.
- Wir kennen den Algorithmus für das Nicht-Leerheitsproblem für Büchi-Automaten.
- Wir sind gespannt darauf zu sehen, was das mit LTL zu tun hat.

Vorlesung 12:
Das Gültigkeitsproblem für LTL

[Literatur: Hoffmann, Lange: Automatentheorie und Logik.]

Der Plan (ganz grob)

Wir wollen das Gültigkeitsproblem für LTL algorithmisch lösen.

Definition 12.1 (Gültigkeitsproblem für LTL)

Das Gültigkeitsproblem für LTL ist folgendes Entscheidungsproblem.

gegeben:	LTL-Formel φ
gefragt:	gilt $\pi \models_{p} \varphi$ für jede Belegungsfolge π ?

Idee:

1. Wir konstruieren einen Büchi-Automaten $B_{\neg\varphi}$ aus der Formel $\neg\varphi$, der alle Belegungsfolgen akzeptiert, die $\neg\varphi$ erfüllen.
2. Nun gilt: φ ist gültig genau dann, wenn $L_\omega(B_{\neg\varphi}) = \emptyset$.
   Also kann man das Gültigkeitsproblem für LTL-Formeln mit Hilfe des Algorithmus für das Leerheitsproblem für Büchi-Automaten lösen.

Der Büchi-Automat für die LTL-Formel

$$\neg\,\mathsf{F}(A \wedge \mathsf{G}(\neg C \rightarrow \mathsf{X}\,\neg B)) \qquad [\equiv \mathsf{G}(A \rightarrow \mathsf{F}(\neg C \wedge \mathsf{X}\,B))]$$

- soll z.B. folgende Belegungsfolgen akzeptieren:
  $\{A, B, C\}\{A, B\}\{B, C\}\{A, B, C\}\{B\}\{B\}\cdots$
  $\{A\}(\ \{B, C\}\,\emptyset\,\{A, B\}\{A, C\}\ )^\omega$
- soll z.B. folgende Belegungsfolgen verwerfen:
  $\{A, B\}\{B\}\{A\}\{A, B, C\}\{A, B\}\,\emptyset^\omega$
  $\{A\}\{A\}\{A\}\{A\}\{A\}^\omega$

Da er mindestens eine Belegungsfolge akzeptiert,
ist $\mathsf{F}(A \wedge \mathsf{G}(\neg C \rightarrow \mathsf{X}\,\neg B))$ keine gültige Formel.

12.1 Konsistente Formelmengen(folgen)

Welche Eigenschaften hat ρ mit $\rho \models_{p} \varphi$?

Dazu schauen wir uns an jedem Punkt von ρ die Formeln an, die dort erfüllt werden.

(Wir betrachten Formeln aus Atomen $\perp$, $\rightarrow$, X und U.)

Definition 12.2 (φ-ähnliche Belegungsfolgen und Mengen(folgen))

Sei φ eine LTL-Formel, und $\rho = \rho_0\rho_1 \ldots$ sei eine Belegungsfolge. Die Teilformelmenge $Z \subseteq \mathsf{Tf}(\varphi)$ mit $Z = \{\alpha \in \mathsf{Tf}(\varphi) \mid \rho \models_{p} \alpha\}$ heißt φ-ähnlich zu ρ ($\rho \overset{\varphi}{\simeq} Z$).

Die Teilformelmengenfolge $Z = Z_0, Z_1, \ldots$ mit $\rho^i \overset{\varphi}{\simeq} Z_i$ für alle $i \geqslant 0$ heißt φ-ähnlich zu ρ ($\rho \overset{\varphi}{\simeq} Z$).

Beispiel:

Sei $\varphi = A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)$ und $\rho = \underbrace{\{A, B\}}_{\rho_0}\underbrace{\{A, B\}}_{\rho_1}\underbrace{\{B\}}_{\rho_2}\underbrace{\{A\}}_{\rho_3}(\underbrace{\{A, B\}}_{\rho_{4+n}})^{\omega}$.

Dann ist

- $Z_0 = \{A,\ B\}$ ist φ-ähnlich zu $\rho\ (= \rho^0)$
- $Z_1 = \{A,\ B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^1
- $Z_2 = \{\neg A,\ B,\ \neg A \,\mathsf{U}\, \neg B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^2
- $Z_3 = \{A,\ \neg B,\ \neg A \,\mathsf{U}\, \neg B\}$ ist φ-ähnlich zu ρ^3
- $Z_{4+n} = \{A,\ B\}$ ist φ-ähnlich zu ρ^{4+n}

Beispiel:

Sei $\varphi = A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)$ und $\rho = \underbrace{\{A, B\}}_{\rho_0}\underbrace{\{A, B\}}_{\rho_1}\underbrace{\{B\}}_{\rho_2}\underbrace{\{A\}}_{\rho_3}(\underbrace{\{A, B\}}_{\rho_{4+n}})^{\omega}$.

Dann ist

- $Z_0 = \{A,\ B\}$ ist φ-ähnlich zu ρ $(= \rho^0)$
- $Z_1 = \{A,\ B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^1
- $Z_2 = \{\neg A,\ B,\ \neg A \,\mathsf{U}\, \neg B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^2
- $Z_3 = \{A,\ \neg B,\ \neg A \,\mathsf{U}\, \neg B\}$ ist φ-ähnlich zu ρ^3
- $Z_{4+n} = \{A,\ B\}$ ist φ-ähnlich zu ρ^{4+n}

Welche lokalen Eigenschaften haben ρ^i und Z_i (für $\alpha \to \beta, \alpha \,\mathsf{U}\, \beta \in \mathsf{Tf}(\varphi)$)?

Beispiel:

Sei $\varphi = A \rightarrow \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)$ und $\rho = \underbrace{\{A, B\}}_{\rho_0} \underbrace{\{A, B\}}_{\rho_1} \underbrace{\{B\}}_{\rho_2} \underbrace{\{A\}}_{\rho_3} (\underbrace{\{A, B\}}_{\rho_{4+n}})^{\omega}$.

Dann ist

- $Z_0 = \{A,\ B\}$ ist φ-ähnlich zu ρ $(= \rho^0)$
- $Z_1 = \{A,\ B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \rightarrow \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^1
- $Z_2 = \{\neg A,\ B,\ \neg A \,\mathsf{U}\, \neg B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \rightarrow \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^2
- $Z_3 = \{A,\ \neg B,\ \neg A \,\mathsf{U}\, \neg B\}$ ist φ-ähnlich zu ρ^3
- $Z_{4+n} = \{A,\ B\}$ ist φ-ähnlich zu ρ^{4+n}

Welche lokalen Eigenschaften haben ρ^i und Z_i (für $\alpha \rightarrow \beta, \alpha \,\mathsf{U}\, \beta \in \mathsf{Tf}(\varphi)$)?

- $\rho^i \not\models_p \bot$ und $\rho^i \models_p \alpha \rightarrow \beta$ gdw. $\rho^i \not\models_p \alpha$ oder $\rho^i \models_p \beta$
- $\bot \notin Z_i$ und $\alpha \rightarrow \beta \in Z_i$ gdw. $\alpha \notin Z_i$ oder $\beta \in Z_i$

Beispiel:

Sei $\varphi = A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)$ und $\rho = \underbrace{\{A, B\}}_{\rho_0}\underbrace{\{A, B\}}_{\rho_1}\underbrace{\{B\}}_{\rho_2}\underbrace{\{A\}}_{\rho_3}(\underbrace{\{A, B\}}_{\rho_{4+n}})^{\omega}$.

Dann ist

- $Z_0 = \{A,\ B\}$ ist φ-ähnlich zu ρ $(= \rho^0)$
- $Z_1 = \{A,\ B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^1
- $Z_2 = \{\neg A,\ B,\ \neg A \,\mathsf{U}\, \neg B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^2
- $Z_3 = \{A,\ \neg B,\ \neg A \,\mathsf{U}\, \neg B\}$ ist φ-ähnlich zu ρ^3
- $Z_{4+n} = \{A,\ B\}$ ist φ-ähnlich zu ρ^{4+n}

Welche lokalen Eigenschaften haben ρ^i und Z_i (für $\alpha \to \beta, \alpha \,\mathsf{U}\, \beta \in \mathsf{Tf}(\varphi)$)?

- $\bot \notin Z_i$ und $\alpha \to \beta \in Z_i$ gdw. $\alpha \notin Z_i$ oder $\beta \in Z_i$
- wenn $\beta \in Z_i$, dann $\alpha \,\mathsf{U}\, \beta \in Z_i$
- wenn $\alpha \,\mathsf{U}\, \beta \in Z_i$, dann $\alpha \in Z_i$ oder $\beta \in Z_i$

Diese Eigenschaften nennen wir *lokale Konsistenz*.

Beispiel:

Sei $\varphi = A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)$ und $\rho = \underbrace{\{A, B\}}_{\rho_0}\underbrace{\{A, B\}}_{\rho_1}\underbrace{\{B\}}_{\rho_2}\underbrace{\{A\}}_{\rho_3}(\underbrace{\{A, B\}}_{\rho_{4+n}})^{\omega}$.

Dann ist

- $Z_0 = \{A,\ B\}$ ist φ-ähnlich zu ρ $(= \rho^0)$
- $Z_1 = \{A,\ B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^1
- $Z_2 = \{\neg A,\ B,\ \neg A \,\mathsf{U}\, \neg B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^2
- $Z_3 = \{A,\ \neg B,\ \neg A \,\mathsf{U}\, \neg B\}$ ist φ-ähnlich zu ρ^3
- $Z_{4+n} = \{A,\ B\}$ ist φ-ähnlich zu ρ^{4+n}

Welche Eigenschaften haben aufeinanderfolgende Z_i und Z_{i+1}?

Beispiel:

Sei $\varphi = A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)$ und $\rho = \underbrace{\{A, B\}}_{\rho_0}\underbrace{\{A, B\}}_{\rho_1}\underbrace{\{B\}}_{\rho_2}\underbrace{\{A\}}_{\rho_3}(\underbrace{\{A, B\}}_{\rho_{4+n}})^\omega$.

Dann ist

- $Z_0 = \{A,\ B\}$ ist φ-ähnlich zu ρ $(= \rho^0)$
- $Z_1 = \{A,\ B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^1
- $Z_2 = \{\neg A,\ B,\ \neg A \,\mathsf{U}\, \neg B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^2
- $Z_3 = \{A,\ \neg B,\ \neg A \,\mathsf{U}\, \neg B\}$ ist φ-ähnlich zu ρ^3
- $Z_{4+n} = \{A,\ B\}$ ist φ-ähnlich zu ρ^{4+n}

Welche Eigenschaften haben aufeinanderfolgende Z_i und Z_{i+1}?

- $\mathsf{X}\,\alpha \in Z_i$ gdw. $\alpha \in Z_{i+1}$, für Teilformeln $\mathsf{X}\,\alpha$ von φ

Beispiel:

Sei $\varphi = A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)$ und $\rho = \underbrace{\{A, B\}}_{\rho_0}\underbrace{\{A, B\}}_{\rho_1}\underbrace{\{B\}}_{\rho_2}\underbrace{\{A\}}_{\rho_3}(\underbrace{\{A, B\}}_{\rho_{4+n}})^{\omega}$.

Dann ist

- $Z_0 = \{A,\ B\}$ ist φ-ähnlich zu ρ $(= \rho^0)$
- $Z_1 = \{A,\ B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^1
- $Z_2 = \{\neg A,\ B,\ \neg A \,\mathsf{U}\, \neg B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$
  ist φ-ähnlich zu ρ^2
- $Z_3 = \{A,\ \neg B,\ \neg A \,\mathsf{U}\, \neg B\}$ ist φ-ähnlich zu ρ^3
- $Z_{4+n} = \{A,\ B\}$ ist φ-ähnlich zu ρ^{4+n}

Welche Eigenschaften haben aufeinanderfolgende Z_i und Z_{i+1}?

- $\mathsf{X}\,\alpha \in Z_i$ gdw. $\alpha \in Z_{i+1}$, für Teilformeln $\mathsf{X}\,\alpha$ von φ
- $\alpha \,\mathsf{U}\, \beta \in Z_i$ gdw. $\beta \in Z_i$ oder $\ulcorner \alpha \in Z_i$ und $\alpha \,\mathsf{U}\, \beta \in Z_{i+1} \urcorner$,
  für $\alpha \,\mathsf{U}\, \beta \in \mathrm{Tf}(\varphi)$

Beispiel:

Sei $\varphi = A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)$ und $\rho = \underbrace{\{A, B\}}_{\rho_0}\underbrace{\{A, B\}}_{\rho_1}\underbrace{\{B\}}_{\rho_2}\underbrace{\{A\}}_{\rho_3}(\underbrace{\{A, B\}}_{\rho_{4+n}})^{\omega}$.

Dann ist

- $Z_0 = \{A,\ B\}$ ist φ-ähnlich zu ρ $(= \rho^0)$
- $Z_1 = \{A,\ B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^1
- $Z_2 = \{\neg A,\ B,\ \neg A \,\mathsf{U}\, \neg B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^2
- $Z_3 = \{A,\ \neg B,\ \neg A \,\mathsf{U}\, \neg B\}$ ist φ-ähnlich zu ρ^3
- $Z_{4+n} = \{A,\ B\}$ ist φ-ähnlich zu ρ^{4+n}

Welche Eigenschaften haben aufeinanderfolgende Z_i und Z_{i+1}?

- $\mathsf{X}\,\alpha \in Z_i$ gdw. $\alpha \in Z_{i+1}$, für Teilformeln $\mathsf{X}\,\alpha$ von φ
- $\alpha \,\mathsf{U}\, \beta \in Z_i$ gdw. $\beta \in Z_i$ oder $\ulcorner \alpha \in Z_i$ und $\alpha \,\mathsf{U}\, \beta \in Z_{i+1} \urcorner$, für $\alpha \,\mathsf{U}\, \beta \in \mathrm{Tf}(\varphi)$

Diese Eigenschaften nennen wir *Nachbar-Konsistenz*.

Beispiel:

Sei $\varphi = A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)$ und $\rho = \underbrace{\{A, B\}}_{\rho_0}\underbrace{\{A, B\}}_{\rho_1}\underbrace{\{B\}}_{\rho_2}\underbrace{\{A\}}_{\rho_3}(\underbrace{\{A, B\}}_{\rho_{4+n}})^{\omega}$.

Dann ist

- $Z_0 = \{A,\ B\}$ ist φ-ähnlich zu ρ $(= \rho^0)$
- $Z_1 = \{A,\ B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^1
- $Z_2 = \{\neg A,\ B,\ \neg A \,\mathsf{U}\, \neg B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^2
- $Z_3 = \{A,\ \neg B,\ \neg A \,\mathsf{U}\, \neg B\}$ ist φ-ähnlich zu ρ^3
- $Z_{4+n} = \{A,\ B\}$ ist φ-ähnlich zu ρ^{4+n}

Welche Eigenschaften fehlen noch?

Beispiel:

Sei $\varphi = A \rightarrow \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)$ und $\rho = \underbrace{\{A, B\}}_{\rho_0}\underbrace{\{A, B\}}_{\rho_1}\underbrace{\{B\}}_{\rho_2}\underbrace{\{A\}}_{\rho_3}(\underbrace{\{A, B\}}_{\rho_{4+n}})^{\omega}$.

Dann ist

- $Z_0 = \{A,\ B\}$ ist φ-ähnlich zu ρ $(= \rho^0)$
- $Z_1 = \{A,\ B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \rightarrow \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^1
- $Z_2 = \{\neg A,\ B,\ \neg A \,\mathsf{U}\, \neg B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \rightarrow \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^2
- $Z_3 = \{A,\ \neg B,\ \neg A \,\mathsf{U}\, \neg B\}$ ist φ-ähnlich zu ρ^3
- $Z_{4+n} = \{A,\ B\}$ ist φ-ähnlich zu ρ^{4+n}

Welche Eigenschaften fehlen noch?

- wenn $\alpha \,\mathsf{U}\, \beta \in Z_i$, dann gibt es ein $j \geqslant i$ mit $\beta \in Z_j$

Beispiel:

Sei $\varphi = A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)$ und $\rho = \underbrace{\{A, B\}}_{\rho_0} \underbrace{\{A, B\}}_{\rho_1} \underbrace{\{B\}}_{\rho_2} \underbrace{\{A\}}_{\rho_3} (\underbrace{\{A, B\}}_{\rho_{4+n}})^\omega$.

Dann ist

- $Z_0 = \{A,\ B\}$ ist φ-ähnlich zu ρ $(= \rho^0)$
- $Z_1 = \{A,\ B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^1
- $Z_2 = \{\neg A,\ B,\ \neg A \,\mathsf{U}\, \neg B,\ \mathsf{X}(\neg A \,\mathsf{U}\, \neg B),\ A \to \mathsf{X}(\neg A \,\mathsf{U}\, \neg B)\}$ ist φ-ähnlich zu ρ^2
- $Z_3 = \{A,\ \neg B,\ \neg A \,\mathsf{U}\, \neg B\}$ ist φ-ähnlich zu ρ^3
- $Z_{4+n} = \{A,\ B\}$ ist φ-ähnlich zu ρ^{4+n}

Welche Eigenschaften fehlen noch?

- wenn $\alpha \,\mathsf{U}\, \beta \in Z_i$, dann gibt es ein $j \geqslant i$ mit $\beta \in Z_j$

Diese Eigenschaft nennen wir *globale Konsistenz*.

Definition 12.3 (lokal konsistente Menge für φ)

Sei φ LTL-Formel und $Z \subseteq \mathsf{Tf}(\varphi)$ eine Menge von Teilformeln von φ.
Z heißt lokal konsistent (für φ), falls gilt:

1. $\bot \notin Z$
2. für alle $\alpha \rightarrow \beta \in \mathsf{Tf}(\varphi)$:
   $\alpha \rightarrow \beta \in Z$ gdw. $\alpha \notin Z$ oder $\beta \in Z$
   (insbesondere für alle $\neg\alpha \in \mathsf{Tf}(\varphi)$: $\neg\alpha \in Z$ gdw. $\alpha \notin Z$)
3. für $\alpha \,\mathsf{U}\, \beta \in \mathsf{Tf}(\varphi)$:
   wenn $\beta \in Z$, dann $\alpha \,\mathsf{U}\, \beta \in Z$ und
   wenn $\alpha \,\mathsf{U}\, \beta \in Z$, dann $\alpha \in Z$ oder $\beta \in Z$

Lokal konsistente Mengen für φ werden Zustände im Büchi-Automaten für φ sein

...

Lemma 12.4

Sei φ eine LTL-Formel, ρ eine Belegungsfolge und Z die Teilformelmenge mit $\rho \overset{\varphi}{\simeq} Z$.
Dann ist Z lokal konsistent für φ.

Definition 12.5 (Nachbar-Konsistenz)

Sei φ eine LTL-Formel
und $L, R \subseteq \mathsf{Tf}(\varphi)$ seien Mengen von Teilformeln von φ.

$L \vdash_{\overline{X}} R$, falls für alle $\mathsf{X}\,\alpha \in \mathsf{Tf}(\varphi)$ gilt: $\quad \mathsf{X}\,\alpha \in L$ gdw. $\alpha \in R$

$L \vdash_{\overline{U}} R$, falls für alle $\alpha\,\mathsf{U}\,\beta \in \mathsf{Tf}(\varphi)$ gilt:
$\alpha\,\mathsf{U}\,\beta \in L$ gdw. $\beta \in L$ oder $\ulcorner \alpha \in L$ und $\alpha\,\mathsf{U}\,\beta \in R \urcorner$

$L \vdash_{\overline{UX}} R$, falls $L \vdash_{\overline{X}} R$ und $L \vdash_{\overline{U}} R$.

$\vdash_{\overline{UX}}$ wird ein Teil der Zustandsübergangsrelation im Büchi-Automaten für φ sein ...

Lemma 12.6

Sei φ eine LTL-Formel, ρ eine Belegungsfolge und $Z = Z_0, Z_1, \ldots$ die Teilformelmengenfolge mit $\rho \overset{\varphi}{\simeq} Z$.
Dann gilt für alle $i \geqslant 0$: $Z_i \vdash_{\overline{UX}} Z_{i+1}$.

Beweis:

$$Z_i \vdash_X Z_{i+1}: \quad \mathsf{X}\,\alpha \in Z_i \;\Leftrightarrow\; \rho^i \models_p \mathsf{X}\,\alpha \qquad (\rho^i \overset{\varphi}{\simeq} Z_i)$$
$$\Leftrightarrow\; \rho^{i+1} \models_p \alpha \qquad (\text{Semantik von X})$$
$$\Leftrightarrow\; \alpha \in Z_{i+1} \qquad (\rho^{i+1} \overset{\varphi}{\simeq} Z_{i+1})$$

$Z_i \vdash_U Z_{i+1}$:

$\alpha\,\mathsf{U}\,\beta \in Z_i$

$$\Leftrightarrow\; \rho^i \models_p \alpha\,\mathsf{U}\,\beta \qquad (\rho^i \overset{\varphi}{\simeq} Z_i)$$
$$\Leftrightarrow\; \rho^i \models_p \beta \vee (\alpha \wedge \mathsf{X}(\alpha\,\mathsf{U}\,\beta)) \qquad (\text{äquivalente Umformung (10.4)})$$
$$\Leftrightarrow\; \rho^i \models_p \beta \text{ oder } (\rho^i \models_p \alpha \;\&\; \rho^{i+1} \models_p \alpha\,\mathsf{U}\,\beta) \qquad (\text{Semantik})$$
$$\Leftrightarrow\; \beta \in Z_i \text{ oder } (\alpha \in Z_i \;\&\; \alpha\,\mathsf{U}\,\beta \in Z_{i+1}) \qquad (\rho^i \overset{\varphi}{\simeq} Z_i \text{ und } \rho^{i+1} \overset{\varphi}{\simeq} Z_{i+1})$$

✓

Definition 12.7 (globale Konsistenz)

Sei φ eine LTL-Formel, und $Z_i \subseteq \mathsf{Tf}(\varphi)$ für $i \geqslant 0$.
Die Folge $Z_0, Z_1, \ldots$ heißt global konsistent, wenn für alle $i \geqslant 0$ gilt:
wenn $\alpha \,\mathsf{U}\, \beta \in Z_i$, dann gibt es ein $j \geqslant i$ mit $\beta \in Z_j$.

Lemma 12.8

Sei φ eine LTL-Formel, ρ eine Belegungsfolge und Z die Teilformelmengenfolge mit $\rho \overset{\varphi}{\simeq} Z$.
Dann ist Z global konsistent.

Beweis:

$$\begin{array}{lll}
\multicolumn{3}{l}{\alpha \,\mathsf{U}\, \beta \in Z_i} \\
\Rightarrow & \rho^i \models_p \alpha \,\mathsf{U}\, \beta & (\rho^i \overset{\varphi}{\simeq} Z_i) \\
\Rightarrow & \text{es gibt ein } j \geqslant i \text{ mit } \rho^j \models_p \beta & (\text{Semantik}) \\
\Rightarrow & \text{es gibt ein } j \geqslant i \text{ mit } \beta \in Z_j & (\rho^j \overset{\varphi}{\simeq} Z_j) \quad \checkmark
\end{array}$$

Lemmas (12.4), (12.6) und (12.8) zeigen, dass alle „guten“ semantischen Eigenschaften von Belegungsfolgen auf ähnliche Teilformelmengenfolgen übertragen werden.

Folgerung 12.9

Sei φ eine LTL-Formel, ρ eine Belegungsfolge, und Z die Teilformelmengenfolge mit $\rho \overset{\varphi}{\simeq} Z$.
Dann ist Z lokal konsistent, nachbar-konsistent und global konsistent.

Sind Teilformelmengenfolgen, die die „guten“ Eigenschaften haben, auch ähnlich zu Belegungsfolgen?

Ja!!

Lemma 12.10

Sei φ eine LTL-Formel,
und $Z = Z_0, Z_1, \ldots$ mit $Z_i \subseteq Tf(\varphi)$ habe folgende Eigenschaften.

1. *Z_i ist lokal konsistent (für alle $i \geqslant 0$),*
2. *$Z_i \vdash_{UX} Z_{i+1}$ (für alle $i \geqslant 0$), und*
3. *Z ist global konsistent.*

Dann ist $\rho \overset{\varphi}{\simeq} Z$ für Belegungsfolge $\rho = \rho_0\rho_1\rho_2\cdots$ mit $\rho_i = Z_i \cap At(\varphi)$ (für alle $i \geqslant 0$).

Beweis:
Sei $\rho = \rho_0\rho_1\rho_2\cdots$ mit $\rho_i = Z_i \cap At(\varphi)$.
Zu zeigen ist: $Z_i = \{\alpha \in \mathsf{Tf}(\varphi) \mid \rho^i \models_p \alpha\}$ f.a. $i \geqslant 0$,
d.h. für alle $\alpha \in \mathsf{Tf}(\varphi)$ und alle $i \geqslant 0$ gilt: $\rho^i \models_p \alpha$ gdw. $\alpha \in Z_i$.
Das machen wir mittels Induktion über den Formelaufbau von α.

IA: für $\perp$ und Atome gilt die Behauptung offensichtlich.

IV: für $\beta, \gamma \in \mathsf{Tf}(\varphi)$ und alle $i \geqslant 0$ gilt:

$$\rho^i \models_p \beta \text{ gdw. } \beta \in Z_i, \text{ und } \rho^i \models_p \gamma \text{ gdw. } \gamma \in Z_i.$$

IS:

$$\begin{array}{llcll}
\alpha = \beta \to \gamma : & \rho^i \models_p \beta \to \gamma & \Leftrightarrow & \rho^i \not\models_p \beta \text{ oder } \rho^i \models_p \gamma & \text{(Semantik)} \\
 & & \Leftrightarrow & \beta \notin Z_i \text{ oder } \gamma \in Z_i & \text{(IV)} \\
 & & \Leftrightarrow & \beta \to \gamma \notin Z_i & (\varphi\text{-Konsistenz von } Z_i) \\
 & & & & \\
\alpha = \mathsf{X}\,\beta : & \rho^i \models_p \mathsf{X}\,\beta & \Leftrightarrow & \rho^{i+1} \models_p \beta & \text{(Semantik)} \\
 & & \Leftrightarrow & \beta \in Z_{i+1} & \text{(IV)} \\
 & & \Leftrightarrow & \mathsf{X}\,\beta \in Z_i & (Z_i \vdash_X Z_{i+1})
\end{array}$$

Bemerkung 1:

$L \mathrel{|\!\!\underline{\;\;}}_{U}\, R$ bedeutet: $\beta \,\mathsf{U}\, \gamma \in L$ gdw. $\gamma \in L$ oder $\ulcorner \beta \in L$ und $\beta \,\mathsf{U}\, \gamma \in R \urcorner$.

Aus $Z_{k-1} \mathrel{|\!\!\underline{\;\;}}_{U} Z_k$ und $\beta \in Z_{k-1}$ und $\gamma \in Z_k$ folgt $\beta \,\mathsf{U}\, \gamma \in Z_k$ und $\beta \,\mathsf{U}\, \gamma \in Z_{k-1}$.

Aus $Z_i \mathrel{|\!\!\underline{\;\;}}_{U} \cdots \mathrel{|\!\!\underline{\;\;}}_{U} Z_k$ und $\beta \in Z_i, \ldots, Z_{k-1}$ und $\gamma \in Z_k$ folgt genauso $\beta \,\mathsf{U}\, \gamma \in Z_i, \ldots, Z_{k-1}$.

Bemerkung 2:

Aus $\beta \,\mathsf{U}\, \gamma \in Z_i$ folgt aus der globalen Konsistenz von Z, dass $\gamma \in Z_q$ für ein $q \geqslant i$.

Sei k das kleinste solche q – d.h. $\gamma \in Z_k$ und $\gamma \notin Z_i, \ldots, Z_{k-1}$.

Aus $Z_i \mathrel{|\!\!\underline{\;\;}}_{U} \cdots \mathrel{|\!\!\underline{\;\;}}_{U} Z_k$ folgt $\beta \in Z_i, \ldots, Z_{k-1}$ (und $\beta \,\mathsf{U}\, \gamma \in Z_i, \ldots, Z_k$).

$[i, k]$ ist Schreibweise für $\{i, i+1, \ldots, k\}$.

$\alpha = \beta \,\mathsf{U}\, \gamma$:

$$
\begin{array}{lll}
\multicolumn{3}{l}{\rho^i \models_{p} \beta \,\mathsf{U}\, \gamma} \\
\text{gdw.} & \exists k \geqslant i : \rho^k \models_{p} \gamma \;\&\; \forall j \in [i, k-1] : \rho^j \models_{p} \beta & \text{(Semantik)} \\
\text{gdw.} & \exists k \geqslant i : \gamma \in Z_k \;\&\; \forall j \in [i, k-1] : \beta \in Z_j & \text{(IV)} \\
\text{gdw.} & \beta \,\mathsf{U}\, \gamma \in Z_i & \text{(Bemerkungen 1 und 2)}
\end{array}
$$

✓

Folgerung (12.9) und Lemma (12.10) können nun zu einem Satz zur Beziehung zwischen Belegungsfolgen und Teilformelmengenfolgen zusammengefasst werden.
In diesem Satz stecken bereits die Konstruktion sowie die Korrektheit und Vollständigkeit des Büchi-Automaten B_φ.

Satz 12.11

Sei φ eine LTL-Formel, ρ eine Belegungsfolge und $Z = Z_0, Z_1, \ldots$ die Teilformelmengenfolge mit $\rho \overset{\varphi}{\simeq} Z$.
Die folgenden Aussagen sind äquivalent.

1. $\rho \models_{p} \varphi$.
2. **2.1** *Jedes Z_i ist lokal konsistent für φ, und*
   2.2 $Z_i \vdash_{UX} Z_{i+1}$ *für alle $i \geqslant 0$, und*
   2.3 *Z ist global konsistent, und*
   2.4 $\varphi \in Z_0$.

12.2 Die Konstruktion des Büchi-Automaten B_φ

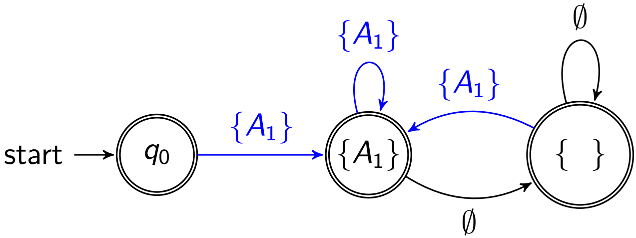
Der verallgemeinerte Büchi-Automat B_φ, der

- die Menge aller Belegungen für $At(\varphi)$ als Alphabet benutzt,
- einen Startzustand hat,
- alle lokal konsistenten Mengen für φ als weitere Zustände hat,
- vom Startzustand in alle Zustände mit φ geht,
- die Zustandsübergänge (1) gemäß $\vdash_{UX}$ und (2) so macht, dass das gelesene Wort (Belegungsfolge) ähnlich den durchlaufenen Zuständen (Teilformelmengenfolge) ist, und
- die Endzustände gemäß der globalen Konsistenz definiert,

akzeptiert alle Belegungsfolgen ρ mit $\rho \models_{p} \varphi$
auf Zustandsfolgen, die (bis auf den Startzustand) ähnlich zu ρ sind.

Beispiel 1: B_{A_1}

$\varphi = A_1$

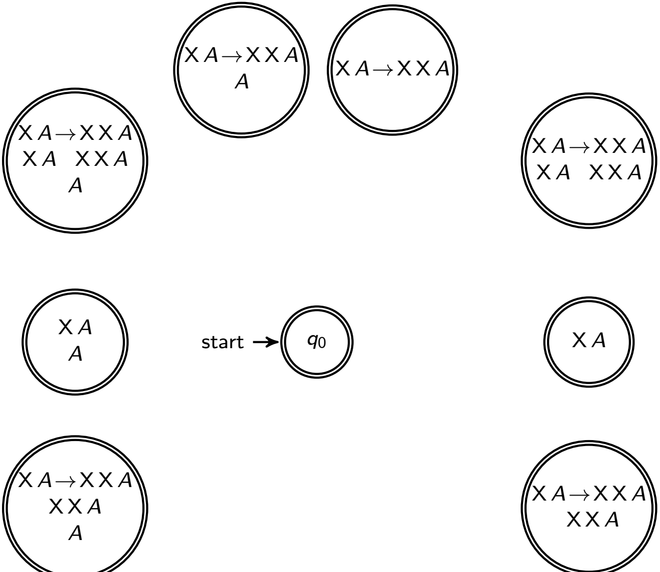


- Zustände $\{q_0\} \cup \{Z \mid Z \subseteq \mathrm{Tf}(A_1)$ ist lokal konsistent$\}$
- Alphabet $\mathcal{P}(At(A_1)) = \mathcal{P}(\{A_1\}) = \{\emptyset, \{A_1\}\}$
- der Startzustand hat alle Zustände q mit $A_1 \in q$ als Nachfolger
- alle Zustandsübergänge benutzen die Belegung des Zielzustandes
- Endzustandsmengen: eine Menge aus allen Zuständen
- $L_\omega(B_{A_1}) = \{\rho \in \{\emptyset, \{A_1\}\}^\omega \mid \rho \models_p A_1\}$

Beispiel 2: $B_{\mathsf{X}A \to \mathsf{X}\mathsf{X}A}$

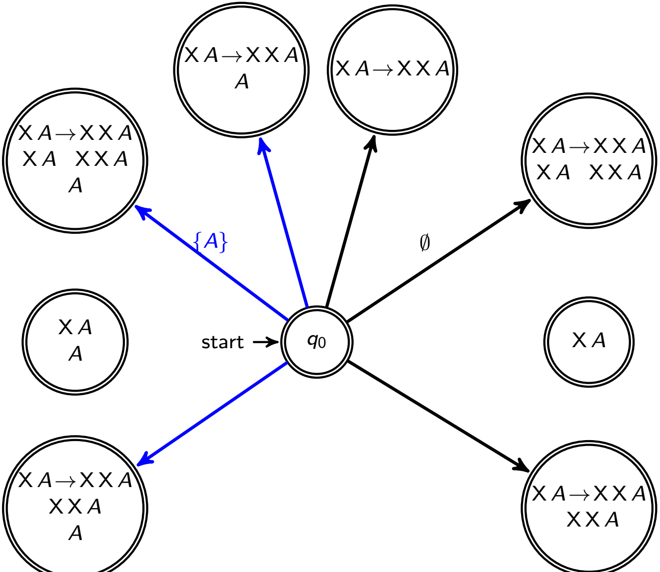
$\varphi = \mathsf{X}A \to \mathsf{X}\mathsf{X}A$

Zustände sind Startzustand q_0 und alle konsistenten Teilmengen von $\mathrm{Tf}(\varphi) = \{\mathsf{X}A \to \mathsf{X}\mathsf{X}A, \mathsf{X}A, \mathsf{X}\mathsf{X}A, A\}$.



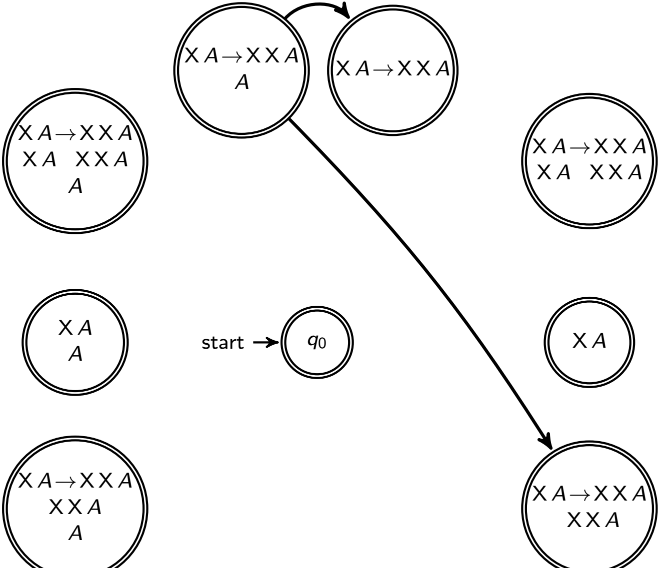
Beispiel 2: $B_{\mathsf{X}\,A \to \mathsf{X}\,\mathsf{X}\,A}$ $\qquad \varphi = \mathsf{X}\,A \to \mathsf{X}\,\mathsf{X}\,A$

Der Startzustand hat alle Zustände mit φ als Nachfolger.
Bei $(L, \mathcal{B}, R) \in \Delta$ gilt stets $\mathcal{B} = R \cap At(\varphi)$.



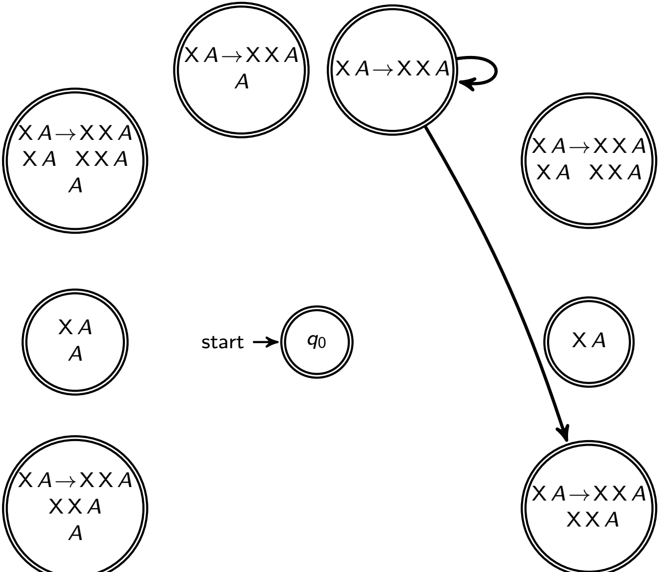
Beispiel 2: $\mathcal{B}_{\mathsf{X}\,A \to \mathsf{X}\,\mathsf{X}\,A}$ $\varphi = \mathsf{X}\,A \to \mathsf{X}\,\mathsf{X}\,A$

Für andere Zustände Z ist Z' Nachfolger, falls $Z \vdash_{UX} Z'$ (hier reicht $Z \vdash_{X} Z'$: $\mathsf{X}\alpha \in Z \Leftrightarrow \alpha \in Z'$).
Bei $(L, \mathcal{B}, R) \in \Delta$ gilt stets $\mathcal{B} = R \cap At(\varphi)$.



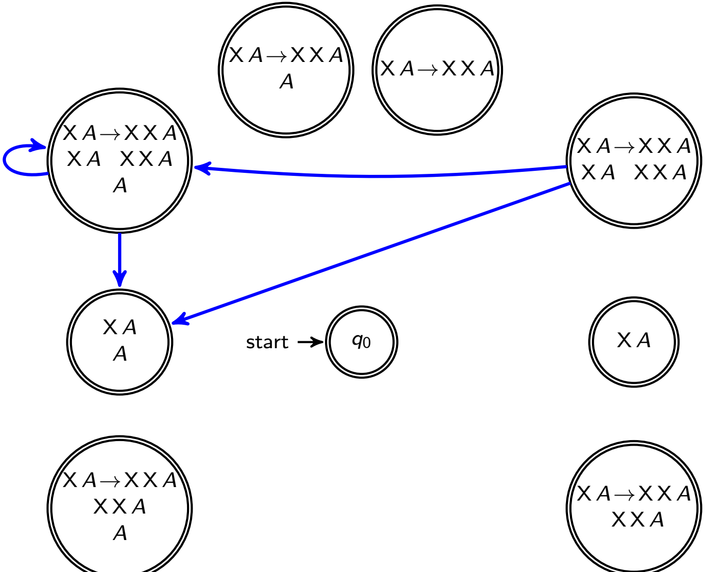
Beispiel 2: $B_{\mathsf{X}\,A \to \mathsf{X}\,\mathsf{X}\,A}$ $\varphi = \mathsf{X}\,A \to \mathsf{X}\,\mathsf{X}\,A$

Für andere Zustände Z ist Z' Nachfolger, falls $Z \vdash_{UX} Z'$ (hier reicht $Z \vdash_{X} Z'$: $\mathsf{X}\alpha \in Z \Leftrightarrow \alpha \in Z'$).
Bei $(L, \mathcal{B}, R) \in \Delta$ gilt stets $\mathcal{B} = R \cap At(\varphi)$.



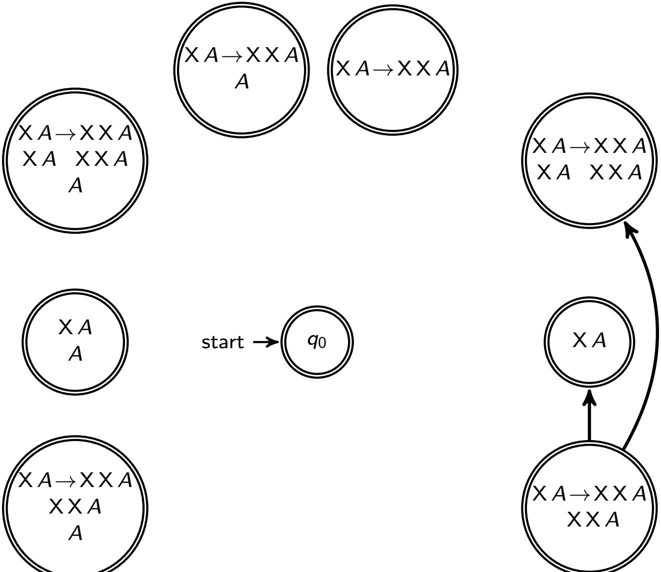
Beispiel 2: $B_{\mathsf{X}\, A \to \mathsf{X}\,\mathsf{X}\, A}$ $\varphi = \mathsf{X}\, A \to \mathsf{X}\,\mathsf{X}\, A$

Für andere Zustände Z ist Z' Nachfolger, falls $Z \vdash_{UX} Z'$ (hier reicht $Z \vdash_{X} Z'$: $X\alpha \in Z \Leftrightarrow \alpha \in Z'$). Bei $(L, \mathcal{B}, R) \in \Delta$ gilt stets $\mathcal{B} = R \cap At(\varphi)$.



Beispiel 2: $B_{\mathsf{X}\,A \rightarrow \mathsf{X}\,\mathsf{X}\,A}$ $\qquad \varphi = \mathsf{X}\,A \rightarrow \mathsf{X}\,\mathsf{X}\,A$

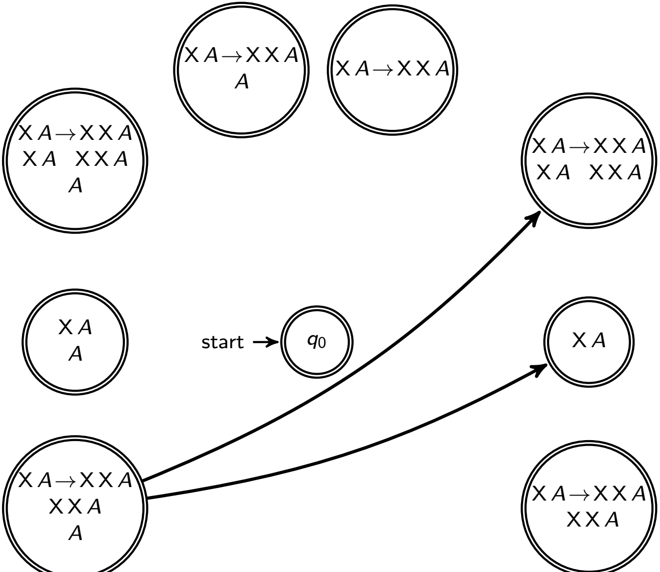
Für andere Zustände Z ist Z' Nachfolger, falls $Z \vdash_{UX} Z'$ (hier reicht $Z \vdash_{X} Z'$: $\mathsf{X}\alpha \in Z \Leftrightarrow \alpha \in Z'$).
Bei $(L, \mathcal{B}, R) \in \Delta$ gilt stets $\mathcal{B} = R \cap At(\varphi)$.



Beispiel 2: $B_{\mathsf{X}\,A\to\mathsf{X}\,\mathsf{X}\,A}$

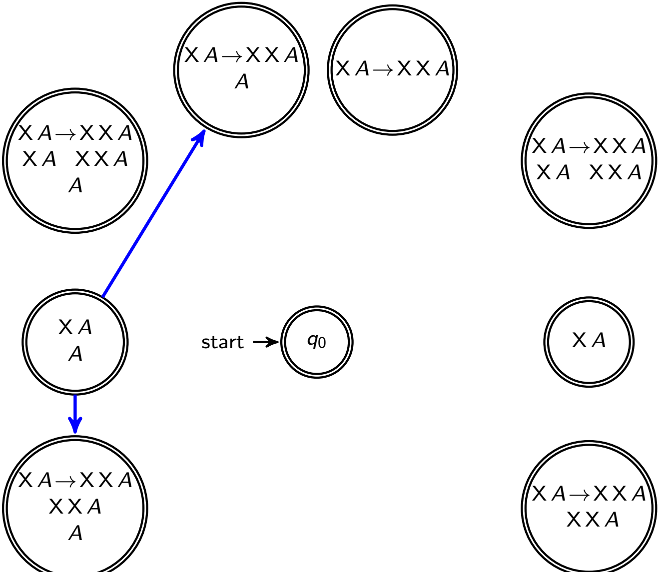
$\varphi = \mathsf{X}\,A \to \mathsf{X}\,\mathsf{X}\,A$

Für andere Zustände Z ist Z' Nachfolger, falls $Z \vdash_{UX} Z'$ (hier reicht $Z \vdash_{X} Z'$: $X\alpha \in Z \Leftrightarrow \alpha \in Z'$).
Bei $(L, \mathcal{B}, R) \in \Delta$ gilt stets $\mathcal{B} = R \cap At(\varphi)$.



Beispiel 2: $B_{\mathsf{X}\,A \to \mathsf{X}\,\mathsf{X}\,A}$ $\varphi = \mathsf{X}\,A \to \mathsf{X}\,\mathsf{X}\,A$

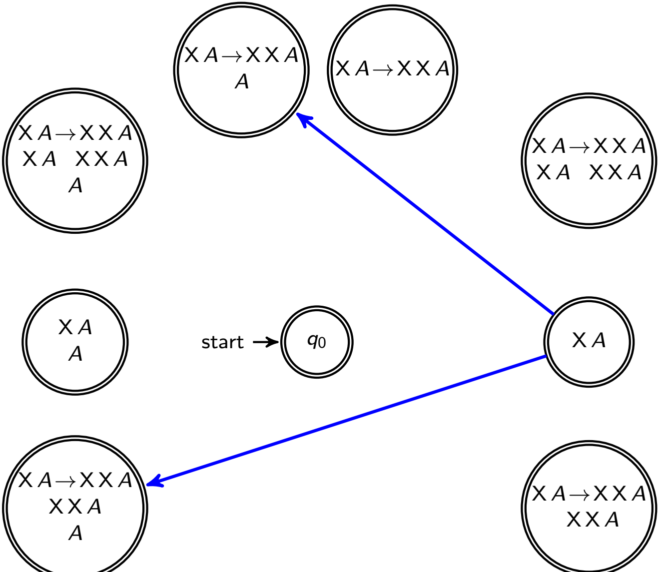
Für andere Zustände Z ist Z' Nachfolger, falls $Z \vdash_{\overline{UX}} Z'$ (hier reicht $Z \vdash_{\overline{X}} Z'$: $\mathsf{X}\alpha \in Z \Leftrightarrow \alpha \in Z'$).
Bei $(L, \mathcal{B}, R) \in \Delta$ gilt stets $\mathcal{B} = R \cap At(\varphi)$.



Beispiel 2: $B_{\mathsf{X}\,A\rightarrow\mathsf{X}\,\mathsf{X}\,A}$

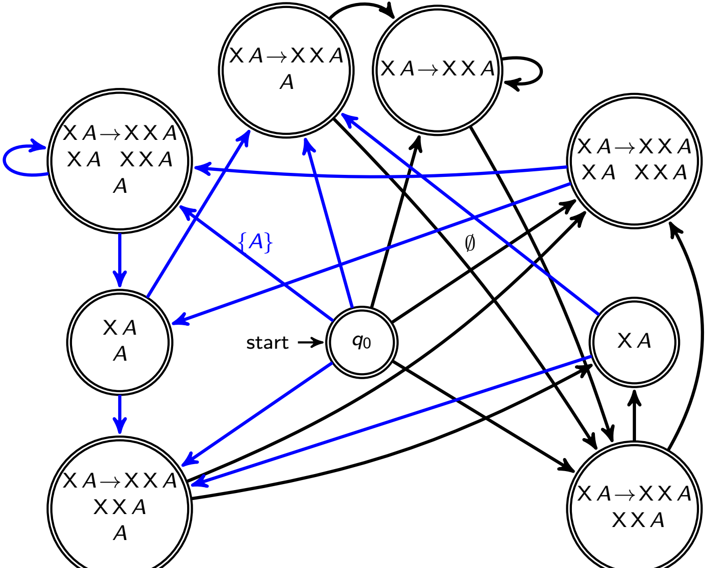
$\varphi = \mathsf{X}\,A \rightarrow \mathsf{X}\,\mathsf{X}\,A$

Für andere Zustände Z ist Z' Nachfolger, falls $Z \vdash_{UX} Z'$ (hier reicht $Z \vdash_{X} Z'$: $\mathsf{X}\alpha \in Z \Leftrightarrow \alpha \in Z'$).
Bei $(L, \mathcal{B}, R) \in \Delta$ gilt stets $\mathcal{B} = R \cap At(\varphi)$.



Beispiel 2: $B_{\mathsf{X}\,A \to \mathsf{X}\,\mathsf{X}\,A}$ $\varphi = \mathsf{X}\,A \to \mathsf{X}\,\mathsf{X}\,A$

Die Menge von Endzustandsmengen ist die Menge mit der Menge aller Zustände als einzigem Element (da kein U in der Formel vorkommt).



Beispiel 3: $B_{\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A))}$ $\qquad \varphi = \neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)) \equiv \mathsf{G}\,\mathsf{F}\,A$

Jeder Zustand $Z \subseteq \mathrm{Tf}(\varphi)$ muss folgende Eigenschaften erfüllen.

1. $\top \in Z$, da $\bot \notin Z$ und $\top \in \mathrm{Tf}(\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)))$
2. entweder $\top \,\mathsf{U}\, A \in Z$ oder $\neg(\top \,\mathsf{U}\, A) \in Z$
3. entweder $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \in Z$ oder $\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)) \in Z$
4. wenn $A \in Z$, dann $\top \,\mathsf{U}\, A \in Z$ und damit $\neg(\top \,\mathsf{U}\, A) \notin Z$
5. wenn $\neg(\top \,\mathsf{U}\, A) \in Z$, dann $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \in Z$ und $\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)) \notin Z$

Es ergeben sich folgende 5 konsistente Mengen für Zustände von $B_{\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A))}$:

	$\top$	A	$[\neg](\top \,\mathsf{U}\, A)$	$[\neg](\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A))$	
q_1:	$\top$		$\top \,\mathsf{U}\, A$	$\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)$	✓
q_2:	$\top$		$\top \,\mathsf{U}\, A$	$\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A))$	✓
	$\top$		$\neg(\top \,\mathsf{U}\, A)$	$\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A))$	geht nicht
q_3:	$\top$		$\neg(\top \,\mathsf{U}\, A)$	$\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)$	✓
q_4:	$\top$	A	$\top \,\mathsf{U}\, A$	$\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)$	✓
q_5:	$\top$	A	$\top \,\mathsf{U}\, A$	$\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A))$	✓
	$\top$	A	$\neg(\top \,\mathsf{U}\, A)$	$\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A))$	geht nicht
	$\top$	A	$\neg(\top \,\mathsf{U}\, A)$	$\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)$	geht nicht

Beispiel 3: $B_{\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A))}$ $\varphi = \neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)) \equiv \mathsf{G}\,\mathsf{F}\, A$

Die Zustandsmenge von B_φ ist $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$.

Für jede Teilformel $\alpha \,\mathsf{U}\, \beta$ von φ gibt es eine Menge von Endzuständen.

1. $$F_{\top \,\mathsf{U}\, A} = \{Z \mid \text{ wenn } \top \,\mathsf{U}\, A \in Z, \text{ dann } A \in Z\} = \{q_3, q_4, q_5\}$$

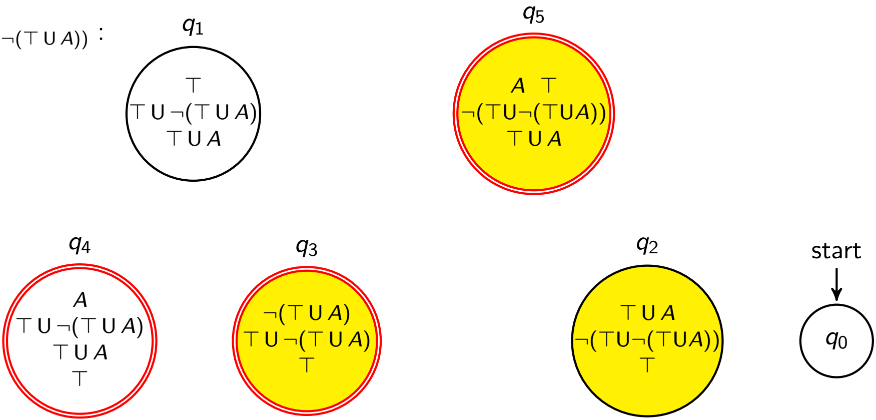
2. $$F_{\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)} = \{Z \mid \text{ wenn } \top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \in Z, \text{ dann } \neg(\top \,\mathsf{U}\, A) \in Z\} = \{q_2, q_3, q_5\}$$

Damit ergibt sich als Menge von Endzustandsmengen von B_φ

$$F = \{Q, F_{\top \,\mathsf{U}\, A}, F_{\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)}\} \quad .$$

Das Alphabet $\Sigma = \{\emptyset, \{A\}\}$ ist die Menge aller Belegungen für das Atom A.

$B_{\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A))}$:



Die Zustandsübergänge in Δ müssen folgende Bedingungen erfüllen.

1. wenn $(Z, \mathcal{B}, Z') \in \Delta$, dann $\mathcal{B} = Z' \cap At(\varphi)$

2. wenn $(q_0, \mathcal{B}, Z') \in \Delta$, dann $\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)) \in Z'$

3. für $(Z, \mathcal{B}, Z') \in \Delta$ gilt $Z \vdash_{\overline{UX}} Z'$ (hier reicht $Z \vdash_{\overline{U}} Z'$, da kein X in φ vorkommt):

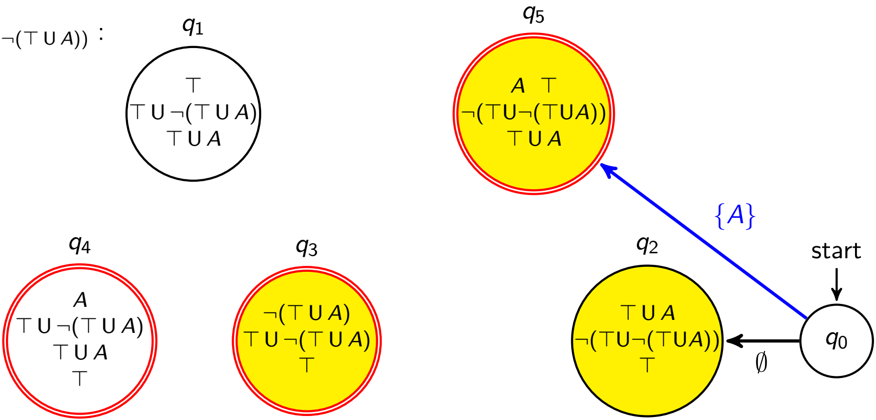
wenn $\top \,\mathsf{U}\, A \in Z$, dann $A \in Z$ oder ($\top \in Z$ und) $\top \,\mathsf{U}\, A \in Z'$

wenn $\top \,\mathsf{U}\, A \notin Z$, dann $A \notin Z$ und ($\top \notin Z$ oder) $\top \,\mathsf{U}\, A \notin Z'$

wenn $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \in Z$, dann $\neg(\top \,\mathsf{U}\, A) \in Z$ oder $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \in Z'$

wenn $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \notin Z$, dann $\neg(\top \,\mathsf{U}\, A) \notin Z$ und $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \notin Z'$

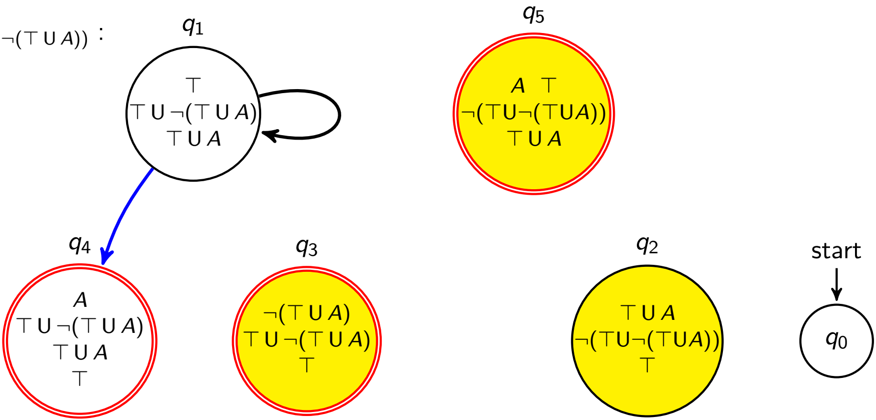
$B_{\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A))}$:



Die Zustandsübergänge in Δ müssen folgende Bedingungen erfüllen.

1. wenn $(Z, \mathcal{B}, Z') \in \Delta$, dann $\mathcal{B} = Z' \cap At(\varphi)$
2. wenn $(q_0, \mathcal{B}, Z') \in \Delta$, dann $\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)) \in Z'$
3. für $(Z, \mathcal{B}, Z') \in \Delta$ gilt $Z \models_{UX} Z'$ (hier reicht $Z \models_{U} Z'$, da kein X in φ vorkommt):
   wenn $\top \,\mathsf{U}\, A \in Z$, dann $A \in Z$ oder $(\top \in Z$ und$)$ $\top \,\mathsf{U}\, A \in Z'$
   wenn $\top \,\mathsf{U}\, A \notin Z$, dann $A \notin Z$ und $(\top \notin Z$ oder$)$ $\top \,\mathsf{U}\, A \notin Z'$
   wenn $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \in Z$, dann $\neg(\top \,\mathsf{U}\, A) \in Z$ oder $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \in Z'$
   wenn $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \notin Z$, dann $\neg(\top \,\mathsf{U}\, A) \notin Z$ und $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \notin Z'$

$B_{\neg(\top \mathsf{U} \neg(\top \mathsf{U} A))}$:



Die Zustandsübergänge in Δ müssen folgende Bedingungen erfüllen.

1. wenn $(Z, \mathcal{B}, Z') \in \Delta$, dann $\mathcal{B} = Z' \cap At(\varphi)$

2. wenn $(q_0, \mathcal{B}, Z') \in \Delta$, dann $\neg(\top \mathsf{U} \neg(\top \mathsf{U} A)) \in Z'$

3. für $(Z, \mathcal{B}, Z') \in \Delta$ gilt $Z \vdash_{UX} Z'$ (hier reicht $Z \vdash_{U} Z'$, da kein X in φ vorkommt):

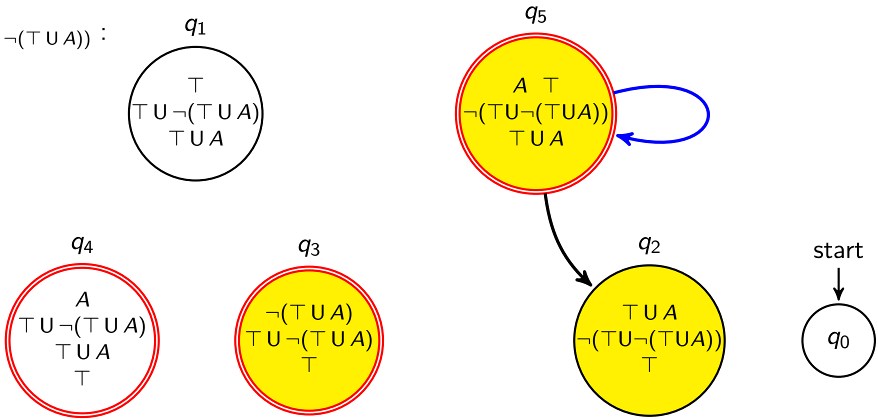
wenn $\top \mathsf{U} A \in Z$, dann $A \in Z$ oder ($\top \in Z$ und) $\top \mathsf{U} A \in Z'$

wenn $\top \mathsf{U} A \notin Z$, dann $A \notin Z$ und ($\top \notin Z$ oder) $\top \mathsf{U} A \notin Z'$

wenn $\top \mathsf{U} \neg(\top \mathsf{U} A) \in Z$, dann $\neg(\top \mathsf{U} A) \in Z$ oder $\top \mathsf{U} \neg(\top \mathsf{U} A) \in Z'$

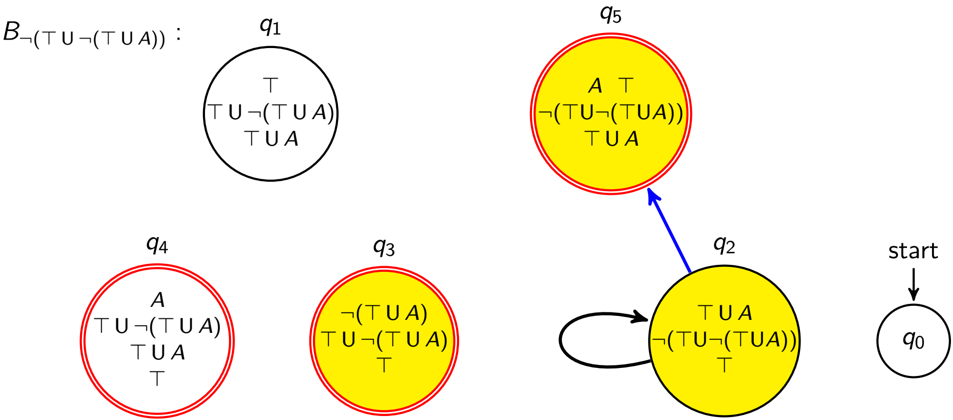
wenn $\top \mathsf{U} \neg(\top \mathsf{U} A) \notin Z$, dann $\neg(\top \mathsf{U} A) \notin Z$ und $\top \mathsf{U} \neg(\top \mathsf{U} A) \notin Z'$

$B_{\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A))}$:



Die Zustandsübergänge in Δ müssen folgende Bedingungen erfüllen.

1. wenn $(Z, \mathcal{B}, Z') \in \Delta$, dann $\mathcal{B} = Z' \cap At(\varphi)$
2. wenn $(q_0, \mathcal{B}, Z') \in \Delta$, dann $\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)) \in Z'$
3. für $(Z, \mathcal{B}, Z') \in \Delta$ gilt $Z \mid_{\overline{UX}} Z'$ (hier reicht $Z \mid_{\overline{U}} Z'$, da kein X in φ vorkommt):
   wenn $\top \,\mathsf{U}\, A \in Z$, dann $A \in Z$ oder ($\top \in Z$ und) $\top \,\mathsf{U}\, A \in Z'$
   wenn $\top \,\mathsf{U}\, A \notin Z$, dann $A \notin Z$ und ($\top \notin Z$ oder) $\top \,\mathsf{U}\, A \notin Z'$
   wenn $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \in Z$, dann $\neg(\top \,\mathsf{U}\, A) \in Z$ oder $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \in Z'$
   wenn $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \notin Z$, dann $\neg(\top \,\mathsf{U}\, A) \notin Z$ und $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \notin Z'$



Die Zustandsübergänge in Δ müssen folgende Bedingungen erfüllen.

1. wenn $(Z, \mathcal{B}, Z') \in \Delta$, dann $\mathcal{B} = Z' \cap At(\varphi)$

2. wenn $(q_0, \mathcal{B}, Z') \in \Delta$, dann $\neg(\top \mathsf{U} \neg(\top \mathsf{U} A)) \in Z'$

3. für $(Z, \mathcal{B}, Z') \in \Delta$ gilt $Z \models_{UX} Z'$ (hier reicht $Z \models_{U} Z'$, da kein X in φ vorkommt):

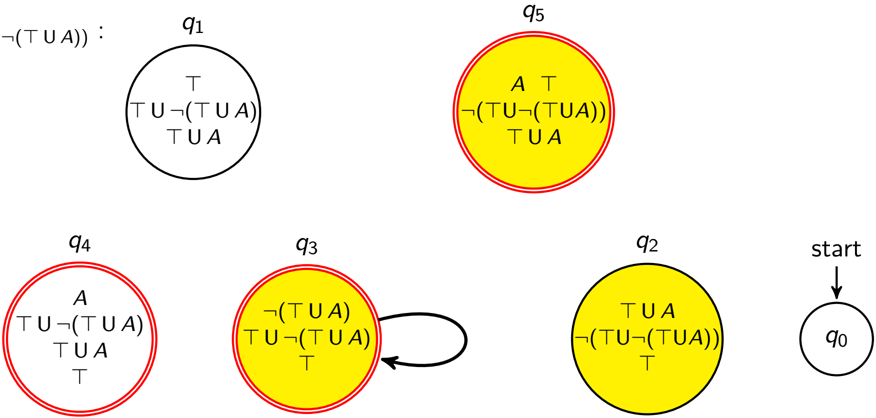
wenn $\top \mathsf{U} A \in Z$, dann $A \in Z$ oder ($\top \in Z$ und) $\top \mathsf{U} A \in Z'$

wenn $\top \mathsf{U} A \notin Z$, dann $A \notin Z$ und ($\top \notin Z$ oder) $\top \mathsf{U} A \notin Z'$

wenn $\top \mathsf{U} \neg(\top \mathsf{U} A) \in Z$, dann $\neg(\top \mathsf{U} A) \in Z$ oder $\top \mathsf{U} \neg(\top \mathsf{U} A) \in Z'$

wenn $\top \mathsf{U} \neg(\top \mathsf{U} A) \notin Z$, dann $\neg(\top \mathsf{U} A) \notin Z$ und $\top \mathsf{U} \neg(\top \mathsf{U} A) \notin Z'$

$B_{\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A))}$:



Die Zustandsübergänge in Δ müssen folgende Bedingungen erfüllen.

1. wenn $(Z, \mathcal{B}, Z') \in \Delta$, dann $\mathcal{B} = Z' \cap At(\varphi)$

2. wenn $(q_0, \mathcal{B}, Z') \in \Delta$, dann $\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)) \in Z'$

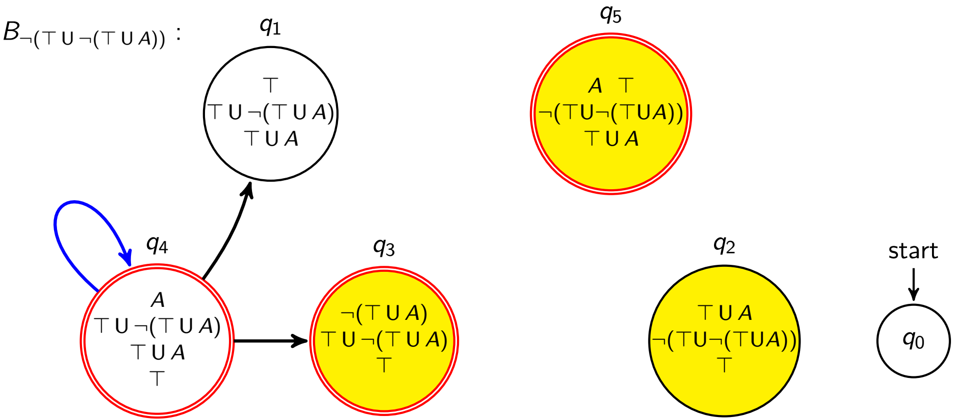
3. für $(Z, \mathcal{B}, Z') \in \Delta$ gilt $Z \left|\!\overline{}_{UX}\right. Z'$ (hier reicht $Z \left|\!\overline{}_{U}\right. Z'$, da kein X in φ vorkommt):

wenn $\top \,\mathsf{U}\, A \in Z$, dann $A \in Z$ oder ($\top \in Z$ und) $\top \,\mathsf{U}\, A \in Z'$

wenn $\top \,\mathsf{U}\, A \notin Z$, dann $A \notin Z$ und ($\top \notin Z$ oder) $\top \,\mathsf{U}\, A \notin Z'$

wenn $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \in Z$, dann $\neg(\top \,\mathsf{U}\, A) \in Z$ oder $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \in Z'$

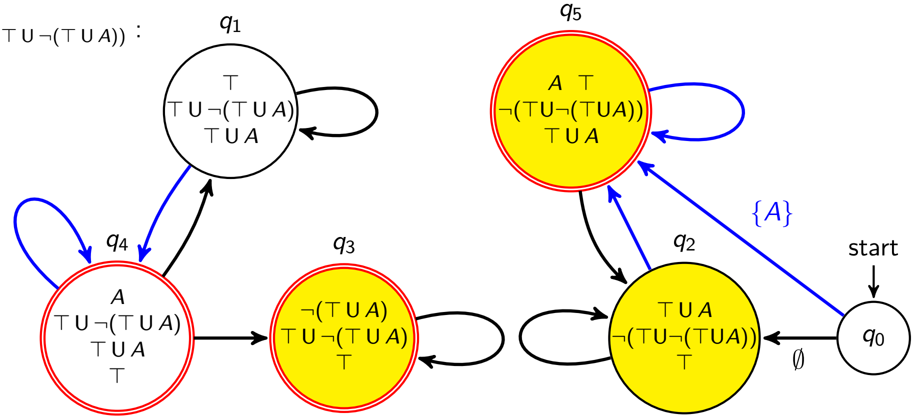
wenn $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \notin Z$, dann $\neg(\top \,\mathsf{U}\, A) \notin Z$ und $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \notin Z'$



Die Zustandsübergänge in Δ müssen folgende Bedingungen erfüllen.

1. wenn $(Z, \mathcal{B}, Z') \in \Delta$, dann $\mathcal{B} = Z' \cap At(\varphi)$
2. wenn $(q_0, \mathcal{B}, Z') \in \Delta$, dann $\neg(\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A)) \in Z'$
3. für $(Z, \mathcal{B}, Z') \in \Delta$ gilt $Z \vdash_{UX} Z'$ (hier reicht $Z \vdash_{U} Z'$, da kein X in φ vorkommt):
   - wenn $\top \,\mathsf{U}\, A \in Z$, dann $A \in Z$ oder ($\top \in Z$ und) $\top \,\mathsf{U}\, A \in Z'$
   - wenn $\top \,\mathsf{U}\, A \notin Z$, dann $A \notin Z$ und ($\top \notin Z$ oder) $\top \,\mathsf{U}\, A \notin Z'$
   - wenn $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \in Z$, dann $\neg(\top \,\mathsf{U}\, A) \in Z$ oder $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \in Z'$
   - wenn $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \notin Z$, dann $\neg(\top \,\mathsf{U}\, A) \notin Z$ und $\top \,\mathsf{U}\, \neg(\top \,\mathsf{U}\, A) \notin Z'$

$B_{\neg(\top \mathsf{U} \neg(\top \mathsf{U} A))}$:



Die Zustandsübergänge in Δ müssen folgende Bedingungen erfüllen.

1. wenn $(Z, \mathcal{B}, Z') \in \Delta$, dann $\mathcal{B} = Z' \cap At(\varphi)$

2. wenn $(q_0, \mathcal{B}, Z') \in \Delta$, dann $\neg(\top \mathsf{U} \neg(\top \mathsf{U} A)) \in Z'$

3. für $(Z, \mathcal{B}, Z') \in \Delta$ gilt $Z \vert\!\underline{\;\;}_{UX}\, Z'$ (hier reicht $Z \vert\!\underline{\;\;}_{U}\, Z'$, da kein X in φ vorkommt):

wenn $\top \mathsf{U} A \in Z$, dann $A \in Z$ oder ($\top \in Z$ und) $\top \mathsf{U} A \in Z'$

wenn $\top \mathsf{U} A \notin Z$, dann $A \notin Z$ und ($\top \notin Z$ oder) $\top \mathsf{U} A \notin Z'$

wenn $\top \mathsf{U} \neg(\top \mathsf{U} A) \in Z$, dann $\neg(\top \mathsf{U} A) \in Z$ oder $\top \mathsf{U} \neg(\top \mathsf{U} A) \in Z'$

wenn $\top \mathsf{U} \neg(\top \mathsf{U} A) \notin Z$, dann $\neg(\top \mathsf{U} A) \notin Z$ und $\top \mathsf{U} \neg(\top \mathsf{U} A) \notin Z'$

Definition 12.12 (durch φ bestimmter Büchi-Automat B_φ)

Sei φ eine LTL-Formel.
Dann ist $B_\varphi = (Q, \Sigma, \Delta, q_0, \mathcal{F})$ der verallgemeinerte Büchi-Automat mit

$$\begin{aligned}
Q &= \{q_0\} \cup \{Z \subseteq \mathrm{Tf}(\varphi) \mid \ Z \text{ ist lokal konsistente Menge für } \varphi\}, \\
\Sigma &= \mathcal{P}(At(\varphi)), \\
\mathcal{F} &= \{F_{\alpha \,\mathsf{U}\, \beta} \mid \alpha \,\mathsf{U}\, \beta \in \mathrm{Tf}(\varphi)\} \cup \{Q\} \text{ mit} \\
&\quad F_{\alpha \,\mathsf{U}\, \beta} = \{q \in Q - \{q_0\} \mid \ \text{wenn } \alpha \,\mathsf{U}\, \beta \in q, \text{ dann } \beta \in q\} \text{ und} \\
\Delta &= \{(q_0, \mathcal{B}, R) \mid \varphi \in R \text{ und } \mathcal{B} = R \cap At(\varphi)\} \\
&\quad \cup \{(L, \mathcal{B}, R) \mid L, R \neq q_0 \text{ und } \mathcal{B} = R \cap At(\varphi) \text{ und } L \vdash_{\underline{UX}} R\}.
\end{aligned}$$

LTL-Gültigkeit und das Leerheitsproblem für Büchi-Automaten

Aus Satz (12.11) folgt ziemlich direkt:

Satz 12.13

Sei φ eine LTL-Formel.
Dann ist $L_\omega(B_\varphi) = \{\rho \in (\mathcal{P}(At(\varphi)))^\omega \mid \rho \models_{\boldsymbol{p}} \varphi\}$.

D.h. B_φ akzeptiert genau die Belegungsfolgen, die φ erfüllen.

Beweisskizze:
Sei $\rho \in L_\omega(B_\varphi)$.
Dann gibt es eine Folge $q_0, Z_0, Z_1, \ldots$ von Zuständen von B_φ, die bei Eingabe ρ durchlaufen wird und die unendlich oft alle Endzustandsmengen von B_φ durchläuft. Aus der Konstruktion von B_φ folgt, dass $Z = Z_0, Z_1, \ldots$ lokal konsistent und nachbar-konsistent ist, und dass $\varphi \in Z_0$. Da Z unendlich oft alle Endzustandsmengen durchläuft, ist Z auch global konsistent. Mit Satz (12.11) folgt $\rho \models_{p} \varphi$.
eine Folge $Z_\rho = Z_0, Z_1, \ldots$ mit $Z_i \subseteq \mathrm{Tf}(\varphi)$, die ähnlich zu ρ ist. Diese Folge ist nach Lemma (12.9) lokal konsistent, nachbar-konsistent und global konsistent ist.
Aus der lokalen Konsistenz und der Nachbar-Konsistenz von Z_ρ sowie der Konstruktion von B_φ folgt, dass $q_0, Z_0, Z_1, \ldots$ eine Folge von Zuständen von B_φ ist, die bei Eingabe ρ durchlaufen wird. Aus der globalen Konsistenz von Z_ρ und der Konstruktion von B_φ folgt, dass diese Zustandsfolge unendlich oft durch jede Endzustandsmenge von B_φ läuft.
Aus der Konstruktion von B_φ folgt schließlich $\varphi \in Z_0$.

Gelte $\rho \models_{p} \varphi$. Dann gibt es die φ-ähnliche Teilformelmengenfolge $Z_\rho = Z_0, Z_1, \ldots$. Für sie gilt nach Satz (12.11): $\varphi \in Z_0$, sie ist lokal konsistent, nachbar-konsistent und global konsistent. Aufgrund der Konstruktion von B_φ ist $q_0, Z_0, Z_1, \ldots$ eine Zustandsfolge, die B_φ bei Eingabe ρ durchlaufen kann. Da sie global konsistent ist, durchläuft sie unendlich oft jede Endzustandsmenge von B_φ. Also folgt $\rho \in L_\omega(B_\varphi)$. ✓

Damit haben wir

Satz 12.14

Sei φ eine LTL-Formel.
Dann ist φ gültig genau dann, wenn $L_\omega(B_{\neg\varphi}) = \emptyset$.

Also lässt sich das Gültigkeitsproblem für LTL mit Hilfe eines Algorithmus für das Leerheitsproblem für Büchi-Automaten lösen ...

12.3

Ein Algorithmus für das Gültigkeitsproblem für LTL

In der letzten Vorlesung (Seite 3.11.21) haben wir einen nichtdeterministischen Algorithmus für das Nicht-Leerheitsproblem für Büchi-Automaten gesehen.

Wir werden ihn jetzt modifizieren, damit er

(1) nichtdeterministisch das Komplement des Gültigkeitsproblem für LTL löst

(2) und dabei möglichst wenig Speicherplatz benötigt.

Algor. für das Komplement des Gültigkeitsproblems für LTL

Eingabe: LTL-Formel φ

$m := \mathit{größeQ}(\neg\varphi)$	(∗ bestimme maximale Pfadlänge ∗)
$q := \mathit{nichtdetF}(\neg\varphi)$	(∗ wähle nichtdet. $q \in F$ ∗)
$x := \mathit{startZustand}(\neg\varphi)$	(∗ x ist der Startzustand ∗)
wiederhole $(m-1)$-mal	(∗ Finde einen Weg von q_0 nach q. ∗)
$v := \mathit{nichtdetQ}(\neg\varphi)$	(∗ wähle nichtdet. $v \in Q$ ∗)
$x := \mathit{nichtdetNachbar}(\neg\varphi, x, v)$	(∗ falls $(x, a, v) \in \Delta$ für ein $a \in \Sigma$, dann $x := v$ ∗
falls $x \neq q$ dann verwirf (und halte)	
$v := \mathit{nichtdetQ}(\neg\varphi)$	(∗ Finde einen Weg von q nach q, ∗)
$x := \mathit{nichtdetNachbar}(\neg\varphi, x, v)$	(∗ der mindestens eine Kante enthält. ∗)
wiederhole $(m-2)$-mal	
$v := \mathit{nichtdetQ}(\neg\varphi)$	
$x := \mathit{nichtdetNachbar}(\neg\varphi, x, v)$	
falls $x = q$	
dann akzeptiere	
sonst verwirf	

Die benutzten Methoden folgen ...

Der verallgemeinerte Büchi-Automat B_ψ

hat $Z \subseteq \mathsf{Tf}(\psi)$ als Zustände und Startzustand q_0.

Daraus kann man wie im Beweis von Lemma (11.11) einen äquivalenten Büchi-Automat $\hat{B}_\psi$ mit Zuständen in $\mathsf{Tf}(\psi) \times \{0, 1, \ldots, \ell\}$.

Es seien für $i = 1, 2, \ldots, \ell$ die Formeln $\gamma_i = \alpha_i \,\mathsf{U}\, \beta_i$ alle U-Teilformeln von ψ.

Die Endzustandsmengen von B_ψ sind $Z, F_1, \ldots, F_\ell$.

Methode *nichtdetQ*(ψ):

(∗ gibt einen nichtdet. gewählten Zustand von $\hat{B}_\psi$ zurück ∗)
wähle nichtdeterministisch $(v, j) \in \mathsf{Tf}(\psi) \times \{0, 1, \ldots, \ell\}$
falls $v \cap At(\psi)$ nicht lokal konsistent für ψ ist: verwirf
return (v, j)

Methode *nichtdetF*(ψ):

(∗ gibt einen nichtdet. gewählten Endzustand von $\hat{B}_\psi$ zurück ∗)
$(v, j) := nichtdetQ(\psi)$
falls $j \geqslant 1$ und $\alpha_j \,\mathsf{U}\, \beta_j \in v$ und $\beta_j \notin v$: verwirf
(∗ für $j = 0$ ist $\mathsf{Tf}(\psi)$ die betrachtete Endzustandsmenge ∗)
return (v, j)

Der verallgemeinerte Büchi-Automat B_ψ

hat $Z \subseteq \mathsf{Tf}(\psi)$ als Zustände und Startzustand q_0.

Daraus kann man wie im Beweis von Lemma (11.11) einen äquivalenten Büchi-Automat $\hat{B}_\psi$ mit Zuständen in $\mathsf{Tf}(\psi) \times \{0, 1, \ldots, \ell\}$.

Es seien für $i = 1, 2, \ldots, \ell$ die Formeln $\gamma_i = \alpha_i \,\mathsf{U}\, \beta_i$ alle U-Teilformeln von ψ.

Die Endzustandsmengen von B_ψ sind $Z, F_1, \ldots, F_\ell$.

Methode *nichtdetNachbar*$(\psi, (x, i), (v, j))$:

(∗ gibt v zurück, falls Übergang von x zu v in $\hat{B}_{\neg\varphi}$ existiert ∗)

wähle nichtdeterministisch $\rho \subseteq At(\psi)$

falls $\rho \neq v \cap At(\psi)$: verwirf

falls x und v nicht nachbar-konsistent sind: verwirf

falls $\ulcorner i = 0$ oder $\gamma_i \in x \Rightarrow \beta_i \in x \urcorner$ und $j \neq (i+1) \bmod \ell$: verwirf

return (v, j)

Methode *größeQ*(ψ):

(∗ gibt eine grobe Anschätzung der Anzahl der Zustände von $\hat{B}_\psi$ zurück ∗)

return $2^{|\mathsf{Tf}(\psi)|} \cdot |\mathsf{Tf}(\psi)|$

Methode *startZustand*(ψ):

(∗ gibt den Startzustand von $\hat{B}_\psi$ zurück ∗)

return $(q_0, 0)$

Analyse des Algorithmus

- Der Algorithmus akzeptiert Eingabe φ, falls $B_{\neg\varphi}$ mindestens eine Belegungsfolge akzeptiert.
  D.h. der Algorithmus akzeptiert Eingabe φ gdw. φ nicht gültig ist.
- Jedes Element von $\mathsf{Tf}(\varphi) \times \{0, 1, \ldots, \ell\}$ benötigt Speicherplatz $O(|\varphi|)$.
  Die benutzten Methoden benötigen ebenfalls Speicherplatz $O(|\varphi|)$.
- Der Algorithmus ist also ein nichtdeterministischer Algorithmus,
  der polynomiellen Speicherplatz benötigt.

Damit haben wir:

das Komplement des Gültigkeitsproblems für LTL ist in NPSPACE.

PSPACE = NPSPACE (nichtdet. Algorithmen kann man in det. Algorithmen umformen, wobei sich der Speicherverbrauch quadriert (Satz von Savitch))

und PSPACE ist unter Komplement abgeschlossen.

Das heißt, der Algorithmus kann in einen deterministischen Alg. für das Gültigkeitsproblem für LTL umgewandelt werden, der polynomiellen Speicherplatz benötigt.

Satz 12.15

Das Gültigkeitsproblem für LTL ist in PSPACE.

Was haben wir in Vorlesung 12 gelernt?

- Zusammenspiel zwischen Belegungsfolgen für LTL-Formeln und ω-Sprachen
- Modellierung von Logik-Problemen durch Büchi-Automaten
- platzsparende algorithmische Umsetzung eines scheinbar exponentiellen Problems

Exkurs und Abschluss von Kapitel 3

Weitere temporale Logiken

Erweiterung von LTL um Pfadquantoren

Computation tree logic (CTL) kennt nur noch Kripke-Semantik (keine Pfad-Semantik auf Belegungsfolgen mehr) und erlaubt Quantifizierung von Pfaden durch Kripke-Modelle mit den Pfad-Quantoren E und A.

Wir geben nur einen kurzen informellen Einblick.

Beispiel

E ... es gibt einen Pfad mit der aktuellen Welt als Startpunkt

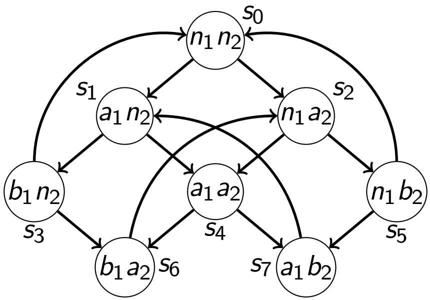
A ... für alle Pfade mit der aktuellen Welt als Startpunkt

$$\mathcal{M}, s_0 \models \mathrm{E}(\mathrm{G}\,\neg b_2)$$

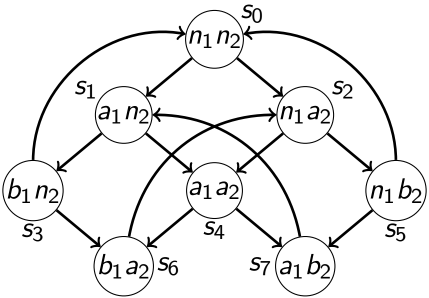
$$\mathcal{M}, s_0 \models \mathrm{E}(\mathrm{G}(\neg b_2 \wedge \mathrm{E}(\mathrm{F}\, b_2)))$$

$$\mathcal{M}, s_0 \models \neg\, \mathrm{E}(\mathrm{F}(b_1 \wedge b_2))$$

$$\mathcal{M}, s_0 \models \mathrm{A}(\mathrm{G}(\neg b_1 \vee \neg b_2))$$

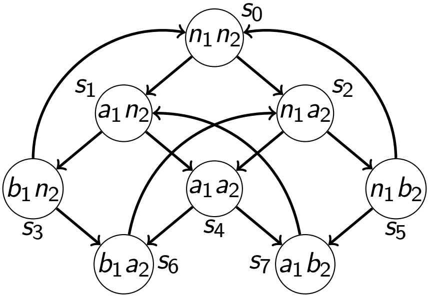


Pfad-Quantoren



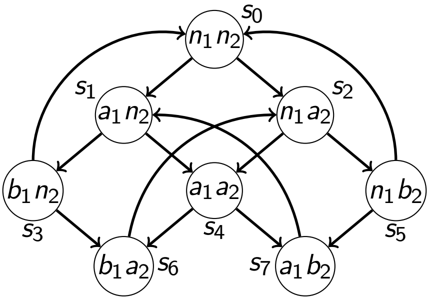
- Sicherheit: $\mathrm{AG}\ \neg(b_1 \wedge b_2)$
- Lebendigkeit: $\mathrm{AG}(a_1 \rightarrow \mathrm{AF}b_1)$
- kein Blockieren: $\mathrm{AG}(n_1 \rightarrow \mathrm{EX}a_1)$
- Flexibilität: $\mathrm{EF}(b_1 \wedge \mathrm{E}[b_1\,\mathrm{U}(\neg b_1 \wedge \mathrm{E}[\neg b_2\,\mathrm{U}\,b_1])])$

Pfad-Quantoren



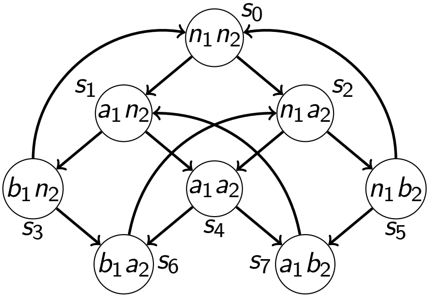
- Sicherheit: AG $\neg(b_1 \wedge b_2)$
- Lebendigkeit: AG$(a_1 \rightarrow \text{AF}b_1)$
- kein Blockieren: AG$(n_1 \rightarrow \text{EX}a_1)$
- Flexibilität: EF$(b_1 \wedge \text{E}[b_1 \,\text{U}(\neg b_1 \wedge \text{E}[\neg b_2 \,\text{U}\, b_1])])$

Pfad-Quantoren



- Sicherheit: AG $\neg(b_1 \wedge b_2)$
- Lebendigkeit: AG$(a_1 \rightarrow \text{AF}b_1)$
- kein Blockieren: AG$(n_1 \rightarrow \text{EX}a_1)$
- Flexibilität: EF$(b_1 \wedge \text{E}[b_1\,\text{U}(\neg b_1 \wedge \text{E}[\neg b_2\,\text{U}\,b_1])])$

Pfad-Quantoren



- Sicherheit: $AG\, \neg(b_1 \wedge b_2)$
- Lebendigkeit: $AG(a_1 \rightarrow AF b_1)$
- kein Blockieren: $AG(n_1 \rightarrow EX a_1)$
- Flexibilität: $EF(b_1 \wedge E[b_1\, U(\neg b_1 \wedge E[\neg b_2\, U\, b_1])])$

Temporale Logiken

LTL: linear-time temporal logic

- Formeln mit temporalen Operatoren, ohne Pfad-Quantoren
$$\mathsf{F}(\mathsf{G}\,x \to y\,\mathsf{U}\,z)$$
- gültige Formel wird in jedem Modell auf *jedem* Pfad erfüllt
- Modell erfüllt Formel,

  wenn sie auf *jedem* Pfad aus s durchs Modell erfüllt wird

CTL: computation tree logic, branching-time temporal logic

- Formeln mit kombinierten Pfad-Quantoren und temporalen Operatoren
$$\mathsf{AF}(\mathsf{EG}x \to \mathsf{A}(y\,\mathsf{U}\,z))$$

CTL^{+}:

- Formeln mit alternierenden Pfad-Quantoren und temporalen Operatoren
$$\mathsf{E}(\mathsf{G}\,x \to (y\,\mathsf{U}(\mathsf{A}(x \wedge \mathsf{F}\,z))))$$

CTL*:

- Formeln mit Pfad-Quantoren und temporalen Operatoren
$$\mathsf{A}(\mathsf{F}\,x \wedge \mathsf{E}(\mathsf{G}\,\mathsf{F}\,x \to (y\,\mathsf{U}\,z)))$$

Komplexitätsresultate

	LTL	CTL	CTL^{+}	$CTL^{\star}$
Erfüllbarkeit / Gültigkeit	PSPACE [SC85]	EXPTIME [FL79] [Pr80]	EEXPTIME [JL03]	EEXPTIME [VS85]
Formelauswertung	PSPACE [SC85]	P [CES86] [Sc02]	Δ_2^{p} [LMS01]	PSPACE [CES86]

$$
\begin{tikzcd}
 & \mathrm{NP} \arrow[dr] & & & & \\
\mathrm{P} \arrow[ur] \arrow[dr] & & \Delta_2^{p} \arrow[r] & \mathrm{PSPACE} \arrow[r] & \mathrm{EXPTIME} \arrow[r] & \mathrm{EEXPTIME} \\
 & \mathrm{coNP} \arrow[ur] & & & &
\end{tikzcd}
$$

P	PSPACE	EXPTIME	EEXPTIME
$\bigcup_k O(n^k)$ Zeit	$\bigcup_k O(n^k)$ Platz	$\bigcup_k O(2^{n^k})$ Zeit	$\bigcup_k O(2^{2^{n^k}})$ Zeit

Komplexitätsresultate

	LTL	CTL	CTL$^+$	CTL*
Erfüllbarkeit / Gültigkeit	PSPACE [SC85]	EXPTIME [FL79] [Pr80]	EEXPTIME [JL03]	EEXPTIME [VS85]
Formelauswertung	PSPACE [SC85]	P [CES86] [Sc02]	Δ_2^p [LMS01]	PSPACE [CES86]

Literatur:

[FL79] Fischer, Ladner 1979
[Pr80] Pratt 1980
[VS85] Vardi, Stockmeyer 1985
[SC85] Sistla, Clarke 1985
[CES86] Clarke, Emerson, Sistla 1986
[Sc02] Schnoebelen 2002
[LMS01] Laroussinie, Markey, Schnoebelen 2001
[JL03] Johannsen, Lange 2003

Was haben wir in Teil 3 gelernt?

- Wir kennen LTL-Formeln mit Pfad-Semantik und Kripke-Semantik.
- Wir haben einen Eindruck von Tableau-Kalkül und Frege-Kalkül für LTL gewonnen.
- Wir kennen Büchi-Automaten und wissen, wie sie ω-Sprachen aus unendlichen Wörtern entscheiden.
- Wir wissen, wie man LTL-Formeln in Büchi-Automaten umwandeln kann, die genau die erfüllenden Berechnungspfade der jeweiligen Formel akzeptieren.
- Wir haben einen Eindruck davon, wie man platzeffizient von einer LTL-Formel entscheiden kann, ob sie gültig ist.