

Wissensrepräsentation

Prof. Dr. Nebel, Dr. Wölfl
M. Helmert, M. Ragni
WS 2005/2006

Universität Freiburg
Institut für Informatik

Projekt P1

Abgabe: Freitag, 20. Januar 2006

Abgabe per E-Mail an helmert@informatik.uni-freiburg.de

Ziel dieses Projekt ist es, den Tableau-Algorithmus für das Schlussfolgern in der Modallogik K zu implementieren. Dies geschieht in vier Schritten:

- Schlussfolgern in Aussagenlogik
- Modaloperatoren
- Gegenbeispiele
- Graphische Tableau-Beweise

Hinweis: Bei den Projektaufgaben geht es um die Implementierung, nicht so sehr um die theoretischen Aspekte. Wenn Sie an bestimmten Stellen nicht genau wissen, wie die Algorithmen und Definitionen aus der Vorlesung in die Praxis umgesetzt werden, können Sie daher gerne Fragen stellen — entweder per E-Mail oder in der Übungsgruppe.

Formeln

Formeln werden durch die folgende einfach gehaltene Grammatik erzeugt:

$S \longrightarrow \mathbf{a} \mid \mathbf{b} \mid \dots \mid \mathbf{z}$	für Aussagevariablen
$S \longrightarrow (S + S)$	für Disjunktionen
$S \longrightarrow (S * S)$	für Konjunktionen
$S \longrightarrow (S \rightarrow S)$	für Implikationen
$S \longrightarrow \sim S$	für Negationen
$S \longrightarrow \mathbf{B} S \mid \mathbf{D} S$	für Box und Diamant

Terminierungskriterien

Um die Terminierung der Tableau-Methode zu garantieren, kann die Expansion des Tableaus anhand der folgenden Regeln eingeschränkt werden:

- **Duplizierte Fakten bei verzweigenden Regeln:** Eine Regel, die das Tableau verzweigt (z.B. für $w \models \varphi \vee \psi$) wird nur auf Äste angewandt, die *keine* der Konsequenzen der Regel (hier also $w \models \varphi$ und $w \models \psi$) enthalten.
- **Duplizierte Fakten bei nicht verzweigenden Regeln:** Eine Regel, die das Tableau nicht verzweigt (z.B. für $w \models \varphi \wedge \psi$) wird nur auf Äste angewandt, die mindestens eine Konsequenz der Regel (hier $w \models \varphi$ und $w \models \psi$) noch nicht enthalten.
- **Einführen neuer Welten:** Eine Regel, die eine neue Welt v einführt (z.B. für $w \models \diamond\varphi$), wird auf jedem Ast *nur einmal* angewandt. (Im Beispiel würde also auf jedem Ast, der $w \models \diamond\varphi$ enthält, *eine* neue Welt v und ein neuer Knoten $v \models \varphi$ eingeführt.)

Aufgabe P1.1 (Schlussfolgern in Aussagenlogik, 2 Punkte)

Implementieren Sie die Tableaumethode zur Überprüfung der Gültigkeit einer aussagenlogischen Formel.

Eingabe

Ihr Programm soll wie folgt ausgeführt werden:¹

```
# tableau <formula>
```

Die Eingabeformel `formula` soll dabei *keine* Box- oder Diamant-Operatoren beinhalten. Wir beschränken uns also für diese Teilaufgabe auf den Teil der Modallogik **K**, der der klassischen Aussagenlogik entspricht.

Zum Beispiel können Sie das folgende Kommando verwenden, um die Regel *Modus Ponens* zu validieren:²

```
# tableau "((p->q)*p)->q"
```

Ausgabe

Die erste Zeile der Programmausgabe soll „**K-valid**.“ lauten, wenn die Eingabeformel **K**-gültig ist. Ansonsten soll sie „**Not K-valid**.“ lauten.

Nach dieser ersten Zeile soll jede folgende Zeile einen Knoten des Tableaus beschreiben. Die genaue Notation kann frei gewählt werden, so lange für jeden Knoten die folgenden Informationen angegeben werden:

- Eine laufende Knotennummer.
- Die Beschriftung des Knotens (von der Art $w \models \varphi$, von der Art $w \not\models \varphi$ oder von der Art wRv).³
- Für innere Knoten des Tableaus: die laufenden Nummern aller Kinder.
- Für Blätter des Tableaus: ein Hinweis, ob dieser Ast des Tableaus offen oder geschlossen ist.

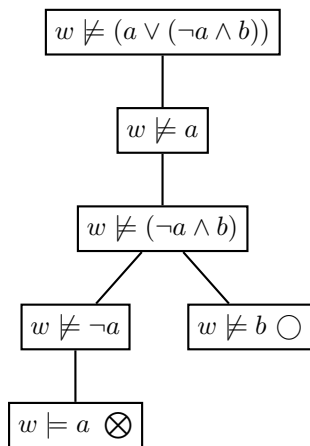
Das folgende Beispiel zeigt einen Tableau-Beweis der Nicht-**K**-Gültigkeit der Formel $(a \vee (\neg a \wedge b))$ zusammen mit der entsprechenden Ausgabe des `tableau`-Programms.⁴

¹Passen Sie die Kommandozeile entsprechend an, wenn Sie eine interpretierte Programmiersprache wie Java verwenden.

²Die Anführungszeichen sind notwendig, um zu verhindern, dass die Symbole `(`, `)`, `>` und `*` von der Shell interpretiert werden.

³Im aussagenlogischen Fall ist die Welt w in allen Knoten derselbe, und es gibt keine Beschriftungen der Art wRv .

⁴Im Tableau sind Blätter geschlossener Äste mit \otimes und Blätter offener Äste mit \circ markiert.



Not K-valid.

- | | |
|--|---------------|
| (N1) not $w \models (a \vee (\neg a * b))$ | [(N2)] |
| (N2) not $w \models a$ | [(N3)] |
| (N3) not $w \models (\neg a * b)$ | [(N4), (N6)] |
| (N4) not $w \models \neg a$ | [(N5)] |
| (N5) $w \models a$ | [closed leaf] |
| (N6) not $w \models b$ | [open leaf] |

Aufgabe P1.2 (Modaloperatoren, 1 Punkt)

Erweitern Sie das Programm aus dem vorigen Aufgabenteil um Modaloperatoren, so dass es beliebige Formeln der Modallogik **K** akzeptiert.

Zum Beispiel soll das folgende Kommando eine Instanz des Axiomschemas *K* validieren:

```
# tableau "(B(a->b)->(Ba->Bb))"
```

Aufgabe P1.3 (Gegenbeispiele, 1 Punkt)

Erweitern Sie das Programm aus dem vorigen Aufgabenteil, so dass es ein Gegenbeispiel ausgibt, wenn die Eingabeformel nicht **K**-gültig ist. Ein Gegenbeispiel ist eine Interpretation, unter der die Formel in einer bestimmten Welt nicht wahr ist.

Ausgabe

Falls die Formel **K**-gültig ist, soll die Ausgabe genauso sein wie bei den vorigen Aufgabenteilen. Ansonsten soll im Anschluss an das Tableau ein Gegenbeispiel ausgegeben werden, das wie folgt notiert ist:

- Zunächst eine Zeile, die die Welten der Interpretation auflistet. In der zuerst genannten Welt soll die Eingabeformel nicht erfüllt sein.
- Als zweites eine Zeile, die die Erreichbarkeitsrelation beschreibt.
- Schließlich für jede Welt eine Zeile, die die Interpretation der Aussagevariablen in dieser Welt beschreibt. Diese Zeilen sollen in der Reihenfolge erscheinen, in der die Welten in der ersten Zeile angegeben wurden.

Zum Beispiel ist die Formel $(\Box a \rightarrow \Box \Box a)$ nicht **K**-gültig, wie in der Vorlesung gezeigt wurde. Das Gegenbeispiel aus der Vorlesung könnte wie folgt ausgegeben werden:

```

Worlds: {u, v, w}
Accessibility Relation: {(u,v), (v,w)}
Truth assignment for u: {a |-> T}
Truth assignment for v: {a |-> T}
Truth assignment for w: {a |-> F}
  
```

Aufgabe P1.4 (Graphische Tableau-Beweise, 0,5 Punkte)

Modifizieren Sie das Programm aus den vorherigen Aufgabenteilen, so dass es optional \LaTeX -Code erzeugt, der den Tableaubeweis graphisch darstellt. (Am einfachsten geht dies mit dem Paket `pst-tree`.)