Einführung in die Modallogik

J.-G. Smaus, S. Wölfl
R. Mattmüller
Sommersemester 2011

Universität Freiburg
Institut für Informatik

Übungsblatt 8

Abgabe: 13. Juli 2011

Aufgabe 8.1 (Tableau-Algorithmus)

In dieser Aufgabe implementieren wir den Tableau-Algorithmus für das Schlussfolgern in der Modallogik **K**.

Die Wahl der Programmiersprache steht Ihnen frei, bei sehr exotischen Sprachen sollten Sie die Wahl aber kurz mit uns absprechen. Abgabe bitte per Mail an `mattmuel@informatik.uni-freiburg.de`.

Ausnahmsweise erlauben und begrüßen wir bei dieser Aufgabe auch die Zusammenarbeit mehrerer Studenten. Die eingereichten Programme müssen die geforderten Ein- und Ausgabeformate verwenden, einige Tests bestehen und **ausreichend kommentiert** sein. Programme, die diesen Anforderungen nicht genügen, werden nicht akzeptiert, aber es besteht die Möglichkeit, innerhalb der Abgabefrist nachzubessern. Daher bitten wir darum, frühzeitig abzugeben, um ausreichend Zeit für Nachbesserungen zu haben.

Im folgenden sind alle notwendigen Schritte angegeben. Die mit einem Stern versehenen Schritte sind nur der Vollständigkeit halber angegeben. Deren Lösung können Sie aus Ihrer Lösung von Aufgabe 2.2 (Blatt 2) übernehmen.

(*a) Implementieren Sie eine interne Repräsentation von $\mathcal{L}_\Box(P)$-Formeln der Gestalt

$$\varphi ::= p \mid \neg\varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \varphi \to \varphi \mid \varphi \leftrightarrow \varphi \mid \Box\varphi \mid \Diamond\varphi \ .$$

(*b) Implementieren Sie eine Prozedur, die in einer gegebenen $\mathcal{L}_\Box(P)$-Formel alle Vorkommen von $\to$ und $\leftrightarrow$ eliminiert, indem $(\varphi \to \psi)$ durch $(\neg\varphi \lor \psi)$ und $(\varphi \leftrightarrow \psi)$ durch $((\neg\varphi \lor \psi) \land (\varphi \lor \neg\psi))$ ersetzt werden.

(*c) Implementieren Sie eine Prozedur zur Herstellung der Negationsnormalform (NNF), die also in einer gegebenen $\mathcal{L}_\Box(P)$-Formel, in der keine $\to$ oder $\leftrightarrow$ mehr vorkommen, alle Negationen möglichst weit nach innen schiebt, indem $\neg(\varphi \lor \psi)$ durch $(\neg\varphi \land \neg\psi)$ ersetzt wird, $\neg(\varphi \land \psi)$ durch $(\neg\varphi \lor \neg\psi)$, $\neg\neg\varphi$ durch φ, sowie $\neg\Box\varphi$ durch $\Diamond\neg\varphi$ und $\neg\Diamond\varphi$ durch $\Box\neg\varphi$.

(d) Schreiben Sie einen Parser, der Formeln in der folgenden Notation einliest und in Ihr internes Format umwandelt (äußerste Klammern sollten optional sein):

$S \longrightarrow$ `p1` $\mid$ `p2` $\mid$ `p3` $\mid\ \dots$ — für Aussagevariablen
$S \longrightarrow$ `(`S `v` S`)` — für Disjunktionen
$S \longrightarrow$ `(`S `&` S`)` — für Konjunktionen
$S \longrightarrow$ `(`S `->` S`)` — für Implikationen
$S \longrightarrow$ `(`S `<->` S`)` — für Biimplikationen
$S \longrightarrow$ `(~`S`)` — für Negationen
$S \longrightarrow$ `(box` S`)` $\mid$ `(dia` S`)` — für Box und Diamant

(e) Implementieren Sie die Tableaumethode zur Überprüfung der **K**-Gültigkeit einer modallogischen Formel (in NNF).

Eingabe:
Ihr Programm soll wie folgt ausgeführt werden:

```
# tableau <formula>
```

Vor Beginn der eigentlichen Tableau-Prozedur sollte die Formel in NNF überführt werden. Das folgende Kommando sollte zum Beispiel eine Instanz des Axiomschemas K validieren:

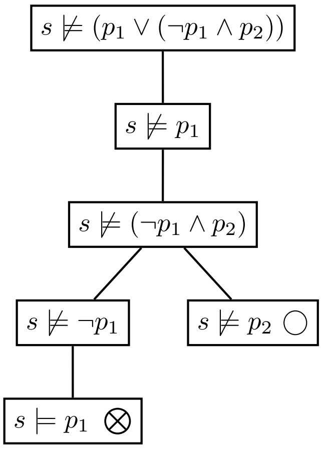
```
# tableau "((box (p1 -> p2)) -> ((box p1) -> (box p2)))"
```

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

Während der Tiefensuche durch das Tableau sollte jeder erzeugte Knoten auf der Standardausgabe ausgegeben werden. 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 $s \models \varphi$, von der Art $s \not\models \varphi$ oder von der Art sRt).
- Die Tiefe im Tableau (durch eine um eine Stufe tiefere Einrückung als beim Elternknoten).
- 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 $(p_1 \vee (\neg p_1 \wedge p_2))$ zusammen mit der entsprechenden Ausgabe des `tableau`-Programms. Im Tableau sind Blätter geschlossener Äste mit $\otimes$ und Blätter offener Äste mit $\bigcirc$ markiert.




```
Not K-valid.
(N1) not s |= (p1 v ((~p1) & p2))
(N2)   not s |= p1
(N3)     not s |= ((~p1) & p2)
(N4)       not s |= (~p1)
(N5)         s |= p1          [closed]
(N6)       not s |= p2        [open]
```

(f) 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.

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 $(\Box p_1 \rightarrow \Box\Box p_1)$ nicht **K**-gültig. Ein Gegenbeispiel könnte wie folgt ausgegeben werden:

```
Worlds: {s, t, u}
Accessibility Relation: {(s,t), (t,u)}
Truth assignment for s: {p1 |-> T}
Truth assignment for t: {p1 |-> T}
Truth assignment for u: {p1 |-> F}
```