Karlsruher Institut für Technologie
Institut für Theoretische Informatik
Prof. Dr. B. Beckert
Dr. des. Vladimir Klebanov

Formale Systeme, WS 2009/2010

Lösungen zum Übungsblatt 4

Dieses Blatt wurde in der Übung am 27.11.2009 besprochen.

Zu Aufgabe 1

(a) Beweis im Sequenzenkalkül.

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{Ax.}
\UnaryInfC{$C \Rightarrow \neg B \wedge \neg A, C$}
\AxiomC{}
\RightLabel{Ax.}
\UnaryInfC{$\neg A, B \Rightarrow B, C$}
\RightLabel{$\neg$rechts}
\UnaryInfC{$\neg A \Rightarrow B, \neg B, C$}
\AxiomC{}
\RightLabel{Ax.}
\UnaryInfC{$\neg A \Rightarrow B, \neg A, C$}
\RightLabel{$\wedge$rechts}
\BinaryInfC{$\neg A \Rightarrow B, \neg B \wedge \neg A, C$}
\RightLabel{$\rightarrow$ rechts}
\UnaryInfC{$\Rightarrow \neg A \rightarrow B, \neg B \wedge \neg A, C$}
\RightLabel{$\rightarrow$ links}
\BinaryInfC{$(\neg A \rightarrow B) \rightarrow C \Rightarrow \neg B \wedge \neg A, C$}
\RightLabel{$\vee$rechts}
\UnaryInfC{$(\neg A \rightarrow B) \rightarrow C \Rightarrow (\neg B \wedge \neg A) \vee C$}
\RightLabel{$\rightarrow$ rechts}
\UnaryInfC{$\Rightarrow ((\neg A \rightarrow B) \rightarrow C) \rightarrow ((\neg B \wedge \neg A) \vee C)$}
\end{prooftree}
$$

(b) Beweis mit Hilfe des Davis-Putnam-Loveland Verfahrens.

Negieren und in KNF bringen:

$$
\begin{aligned}
\neg F \equiv\ & \neg(((\neg A \rightarrow B) \rightarrow C) \rightarrow ((\neg B \wedge \neg A) \vee C)) \equiv \\
& \neg(\neg((\neg A \rightarrow B) \rightarrow C) \vee ((\neg B \wedge \neg A) \vee C)) \equiv \\
& \neg(\neg(\neg(\neg A \rightarrow B) \vee C) \vee ((\neg B \wedge \neg A) \vee C)) \equiv \\
& \neg(\neg(\neg(\neg\neg A \vee B) \vee C) \vee ((\neg B \wedge \neg A) \vee C)) \equiv \\
& \neg\neg(\neg(A \vee B) \vee C) \wedge \neg((\neg B \wedge \neg A) \vee C) \equiv \\
& ((\neg A \wedge \neg B) \vee C) \wedge (\neg(\neg B \wedge \neg A) \wedge \neg C) \equiv \\
& ((\neg A \wedge \neg B) \vee C) \wedge ((\neg\neg B \vee \neg\neg A) \wedge \neg C) \equiv \\
& ((\neg A \vee C) \wedge (\neg B \vee C)) \wedge ((B \vee A) \wedge \neg C) \equiv
\end{aligned}
$$

Klauselform:

$$\{\neg A, C\} \quad \{\neg B, C\} \quad \{B, A\} \quad \{\neg C\}$$

```
function DPLL(S)
   if S is a consistent set of literals
      then return true;
   if S contains an empty clause
      then return false;
   for every unit clause l in S
      S := unit-propagate(l, S);
   l := chooseLiteral(S);
   return DPLL(S_{l←1}) OR DPLL(S_{l←0});
```

Abbildung 1: Zur Wiederholung eine Kurzfassung des Davis-Putnam-Algorithmus

Davis-Putnam-Verfahren tabellarisch:

	$C \leftarrow \mathbf{0}$	$A \leftarrow \mathbf{0}$	$B \leftarrow \mathbf{0}$
$\{\neg A, C\}$	$\{\neg A\}$	$-$	$-$
$\{\neg B, C\}$	$\{\neg B\}$	$\{\neg B\}$	$-$
$\{B, A\}$	$\{B, A\}$	$\{B\}$	$\square$
$\{\neg C\}$	$-$	$-$	$-$

D.h., $\neg F$ ist unerfüllbar, damit ist F allgemeingültig.

Zu Aufgabe 2

(a) Benutze die Signatur $\Sigma = \{H, G, M, Z, D\}$ mit offensichtlicher Bedeutung.

(1) $H \vee G$

(2) $(G \wedge \neg M) \rightarrow Z$

(3) $\neg(Z \wedge D)$

(4) $M \rightarrow \neg D$

(5) $\neg(H \wedge D)$

(6) $Z \rightarrow (\neg G \wedge M)$

(7) $\neg Z \rightarrow D$

(b) Diese Aufgabe ist bereits so angelegt, daß die Umformung in KNF leicht erfolgen kann. Hier wird (für (c)) bereits die korrespondierende Klauselmenge angegeben:

[1]	$\{H, G\}$
[2]	$\{\neg G, M, Z\}$
[3]	$\{\neg Z, \neg D\}$
[4]	$\{\neg M, \neg D\}$
[5]	$\{\neg D, \neg H\}$
[6a]	$\{\neg Z, \neg G\}$
[6b]	$\{\neg Z, M\}$
[7]	$\{Z, D\}$

(c) Diese Klauselmenge S enthält keine Einerklausel, also muß eine Verzweigung stattfinden: $S_{H\leftarrow\mathbf{0}}$ und $S_{H\leftarrow\mathbf{1}}$ (andere wären auch wählbar):

(i) $S_{H\leftarrow\mathbf{0}}$:

$H \leftarrow \mathbf{0}$	$G \leftarrow \mathbf{1}$	$Z \leftarrow \mathbf{0}$	$D \leftarrow \mathbf{1}$	$M \leftarrow \mathbf{1}$
$\{G\}$	–	–	–	–
$\{\neg G, M, Z\}$	$\{M, Z\}$	$\{M\}$	$\{M\}$	–
$\{\neg Z, \neg D\}$	$\{\neg Z, \neg D\}$	–	–	–
$\{\neg M, \neg D\}$	$\{\neg M, \neg D\}$	$\{\neg M, \neg D\}$	$\{\neg M\}$	$\square$
–	–	–	–	–
$\{\neg Z, \neg G\}$	$\{\neg Z\}$	–	–	–
$\{\neg Z, M\}$	$\{\neg Z, M\}$	–	–	–
$\{Z, D\}$	$\{Z, D\}$	$\{D\}$	–	–

(ii) $S_{H\leftarrow\mathbf{1}}$:

$H \leftarrow \mathbf{1}$	$D \leftarrow \mathbf{0}$	$Z \leftarrow \mathbf{1}$	$G \leftarrow \mathbf{0}$	$M \leftarrow \mathbf{1}$
–	–	–	–	–
$\{\neg G, M, Z\}$	$\{\neg G, M, Z\}$	–	–	–
$\{\neg Z, \neg D\}$	–	–	–	–
$\{\neg D\}$	–	–	–	–
$\{\neg Z, \neg G\}$	$\{\neg Z, \neg G\}$	$\{\neg G\}$	–	–
$\{\neg Z, M\}$	$\{\neg Z, M\}$	$\{M\}$	$\{M\}$	–
$\{Z, D\}$	$\{Z\}$	–	–	–

Das Verfahren erreicht nun eine Situation, in der keine Klausel mehr zur Verfügung steht (N.B. Das ist sehr verschieden davon, daß die *leere Klausel* abgeleitet wurde). Das bedeutet, daß jede Belegung, die die Festlegungen des aktuellen Pfades respektiert, eine erfüllende Belegung ist. In diesem Fall ist also die Belegung I mit

$$I(H) = W,\ I(D) = F,\ I(Z) = W,\ I(G) = F,\ I(M) = W$$

eine erfüllende Belegung. Das Davis-Putnam-Verfahren kann mit der Ausgabe „erfüllbar“ terminieren.

Zu Aufgabe 3

(a) $j(f(x), g(x), h(z), k)$ ist ein wohlgeformter Term.

(b) $\forall y \exists p\, p(y)$ ist weder Term noch Formel. In PL1 ist es nicht möglich, über Prädikatensymbole zu quantifizieren ($\exists p$).

(c) $\forall x \forall z\, (g(f(z), f(y)) \to z)$ ist weder Term noch Formel. Implikation zwischen Termen ist nicht wohlgeformt.

(d) $\forall x \exists y\, (p(f(\underset{\substack{\uparrow\\ \text{gebunden}}}{x})) \to q(\overset{\substack{\text{gebunden}\\ \downarrow}}{y}, \underset{\substack{\uparrow\\ \text{frei}}}{z}))$ ist eine Formel. Die Variablenvorkommen sind entsprechend markiert.

Zu Aufgabe 4

1. $\forall x(x \doteq a \lor x \doteq b \lor x \doteq c) \;\land\; \neg(a \doteq b) \land \neg(a \doteq c) \land \neg(b \doteq c)$
2. $\exists x(\mathrm{kills}(x, a))$
3. $\forall x \forall y(\mathrm{kills}(x, y) \to \mathrm{hates}(x, y))$
4. $\forall x(\mathrm{hates}(a, x) \to \neg\,\mathrm{hates}(c, x))$
5. $\forall x \forall y(\mathrm{kills}(x, y) \to \neg\,\mathrm{richer}(x, y))$
6. $\forall x((\neg\,\mathrm{richer}(x, a) \lor \mathrm{hates}(a, x)) \to \mathrm{hates}(b, x))$
7. $\neg \exists x \forall y(\mathrm{hates}(x, y))$

Anmerkung zu Formel 2: Eigentlich gehört aber die Aussage 1 (Stichwort Hausbewohner) noch mit in die Kodierung, so daß die folgende Formel $2'$ die Aussage noch exakter formalisiert:

$$\exists x(\mathrm{kills}(x, a) \land (x \doteq a \lor x \doteq b \lor x \doteq c)$$

Diese wiederum ist äquivalent zu:

$$\mathrm{kills}(a, a) \lor \mathrm{kills}(b, a) \lor \mathrm{kills}(c, a)$$

Allerdings wird diese Präzisierung – da auch Formel 1 zur Verfügung steht – nicht wirklich benötigt. Die Formeln 1 und 2 sind zusammen äquivalent zu 1 und $2'$.