

Theoretische Informatik I

13. Übung

Geben Sie die Lösung der Aufgabe 1 bitte bis zum 27.01.2014 9:00 Uhr ab. (Briefkasten vorm Raum 1/266 oder per eMail an fal@informatik.tu-chemnitz.de, *Betreff: TI1 Hausaufgaben*)

1. Aufgabe: Eine einfache Möglichkeit die Erfüllbarkeit einer Aussagenlogischen Formel zu testen, ist ein *Backtracking* über die in der Formel vorkommenden Variablen durchzuführen. Dieses Vorgehen nach *Davis* und *Putnam* basiert auf der folgenden Überlegung:

Wir nehmen eine Variable x aus der Formel. Nun ersetzen wir jedes Vorkommen von x durch 0 bzw. 1. Dadurch entsteht eine „Restformel“, die wir mit $F|_{x=0}$ bzw. $F|_{x=1}$ bezeichnen. Diese Restformel läßt sich gegebenenfalls noch vereinfachen und außerdem kommt die Variable x darin nicht vor.

Wenn jetzt $F|_{x=0}$ oder $F|_{x=1}$ *erfüllbar* ist, so ist auch F *erfüllbar*. Sollten jedoch sowohl $F|_{x=0}$ als auch $F|_{x=1}$ *nicht erfüllbar* sein, so ist auch F *nicht erfüllbar*. Es gilt also:

$$F \text{ erfüllbar} \iff F|_{x=0} \text{ oder } F|_{x=1} \text{ erfüllbar.}$$

Dieses Vorgehen läßt sich rekursiv auf den Restformeln fortsetzen.

Bei Formeln in k -KNF kann man diese Prozedur besonders günstig implementieren. Wir betrachten im folgenden die beiden 3-KNF Formeln F_1 und F_2 .

$$\begin{aligned} F_1 &= (x \vee y) \wedge (\neg y \vee \neg z \vee a) \wedge (z \vee \neg a \vee \neg x) \wedge (\neg y) \\ F_2 &= (a \vee b) \wedge (\neg a \vee b) \wedge (a \vee \neg b) \wedge (\neg a \vee \neg b \vee c) \end{aligned}$$

- (a) (i) Wie wird die Restformel $F|_{x=0}$ bzw. $F|_{x=1}$ inklusive der Vereinfachung der Formel gebildet?
- (ii) Woran erkennt man, dass eine Formel „offensichtlich“ erfüllbar bzw. unerfüllbar ist? (Dies bildet das Ende der Rekursion.)

Hinweis: Überlegen Sie, wie sich eine Klausel $C = (x_1 \vee x_2 \vee \dots \vee x_k)$ vereinfacht, wenn eine der Variablen auf 0 bzw. 1 gesetzt wird.

- (b) Wenden Sie die *Davis-Putnam-Prozedur* auf die beiden 3-KNF Formeln an.
- (i) Setzen Sie die Variablen in alphabetischer Reihenfolge.
- (ii) Zeichnen Sie den Aufrufbaum. Geben Sie für jeden Aufruf die zugehörige Restformel an.
- (c) Begründen Sie anhand des Aufrufbaumes, der sich bei einer *unerfüllbaren* Formel ergibt, welche Laufzeit dieses Verfahren hat.

2. Aufgabe: Transformieren Sie die folgende Formel mit dem Polynomialzeitalgorithmus der Vorlesung in eine erfüllbarkeitsäquivalente Formel in 3-KNF.

$$F_3 = (x \vee \neg y) \Leftrightarrow (y \wedge z)$$

3. Aufgabe: Versuchen Sie für die folgende Formel in 3-KNF eine erfüllende Belegung der Variablen zu finden.

$$F_4 = (a \vee c \vee d) \wedge (a \vee \neg c \vee d) \wedge (\neg a \vee c \vee d) \wedge (\neg a \vee \neg c \vee d) \wedge \\ (b \vee c \vee \neg d) \wedge (b \vee \neg c \vee \neg d) \wedge (\neg b \vee c \vee \neg d) \wedge (\neg b \vee \neg c \vee \neg d)$$

Verwenden Sie für die Suche die Idee aus der Vorlesung, nach einer Klausel zu verzweigen. Die Formel F enthalte eine Klausel C mit den Literalen x, y und z . Dann gilt:

$$F \text{ ist erfüllbar} \Leftrightarrow F_{|x=1} \text{ oder } F_{|x=0,y=1} \text{ oder } F_{|x=0,y=0,z=1} \text{ ist erfüllbar.}$$

(a) Zeichnen Sie den Aufrufbaum.

(b) Zeigen Sie, dass der Aufrufbaum bei diesem Vorgehen weniger als 2^n Blätter hat.

4. Aufgabe: Betrachten Sie folgenden Algorithmus für das Erfüllbarkeitsproblem von 2-KNF Formeln.

DPT(F)

1. Ist F eine leere Formel:
2. Ausgabe: *erfüllbar*
3. Enthält F eine leere Klausel:
4. Ausgabe: *unerfüllbar*
5. Wähle ein beliebiges Literal l , das in F vorkommt
6. $F' = \text{setTrue}(F, l)$
7. Falls $F' \subset F$: // (F' ist Teilformel von F)
8. DPT(F')
9. Ansonsten:
10. DPT(setTrue($F, \neg l$))

setTrue(F, l)

1. $F' = F$
2. Entferne aus F' alle Klauseln, die l enthalten
3. Entferne $\neg l$ aus allen Klauseln von F'
4. Wenn es in F' eine unit-clause $c = \{u\}$ gibt,
 gehe zu 2. mit $l = u$
5. return(F')

(a) Zeigen Sie, dass der Algorithmus korrekt ist.

Hinweis: Überlegen Sie sich dazu, warum der Teilformeltest in Schritt 7 von DPT ausreicht, um das (bei der Davis-Putnam-Prozedur sonst nötige) Backtracking zu vermeiden.

(b) Welche Laufzeit hat der Algorithmus?