

# Theoretische Informatik I

## 10. Übung

Geben Sie die Lösung der Aufgabe 1 bitte bis zum 16.12.2011 ab. (Briefkasten vorm Raum 1/266 oder per eMail an [fallu@informatik.tu-chemnitz.de](mailto:fallu@informatik.tu-chemnitz.de), *Betreff*: TI1 Hausaufgaben)

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

$$(x \vee \neg y) \iff (y \wedge z)$$

**2. Aufgabe:** Wenden Sie die Davis-Putnam-Prozedur auf die 3-KNF

$$(x \vee y) \wedge (\neg y \vee \neg z \vee a) \wedge (z \vee \neg a \vee \neg x) \wedge (\neg y)$$

- (a) ohne Verwendung der Heuristiken *pure-literal* und *unit-clause*
- (b) mit den beiden Heuristiken

an. Zeichnen Sie jeweils den Aufrufbaum.

**3. Aufgabe:** Betrachten Sie folgenden Algorithmus *DPT* 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( $\text{setTrue}(F, \neg l)$ )

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?