Prof. Dr. A. Goerdt
Julian Pape-Lange

TU Chemnitz
Wintersemester 2019/2020
21.10.2019

Theorie der Programmiersprachen

2. Übung

1. Aufgabe:
Die Formel

$$F = (\neg A_1 \vee A_2) \wedge (\neg A_2 \vee A_3) \wedge (\neg A_3 \vee A_4) \wedge (\neg A_4 \vee A_5) \wedge (\neg A_4 \vee \neg A_5)$$

ist gegeben. Wir wollen mit den Algorithmen aus der Vorlesung bestimmen, ob F erfüllbar ist.
Bestimmen Sie die Anzahl der dafür notwendigen Setzungen

(a) im Polynomialzeitalgorithmus für 2-KNF und

(b) im Linearzeitalgorithmus für 2-KNF.

2. Aufgabe:
Zeigen Sie, dass es zu jeder 2-KNF F mit Variablen $A_1, A_2, \ldots, A_n$ eine andere 2-KNF G gibt, sodass

- F genau dann erfüllbar ist, wenn G erfüllbar ist,
- G keine Klauseln der Form $X \vee X$ enthält,
- G nur 2 Klauseln mehr als F enthält,
- G nur 2 Variablen mehr als F enthält und
- F für jede erfüllende Belegung von G auch erfüllt ist.

3. Aufgabe:
Die 2-KNF F ohne einelementige Klauseln ist, wie im Linearzeitalgorithmus für die 2-KNF aus der Vorlesung, in Adjazenzlistendarstellung gegeben. Jede Klausel $A_i \vee A_j$ wird also sowohl durch einen Eintrag in der Adjazenzliste von A_i als auch durch einen Eintrag in der Adjazenzliste von A_j repräsentiert. Es fehlen jedoch die Pointer zwischen den beiden Kopien der Klausel.
Geben Sie einen Algorithmus an, der zu jedem Eintrag den zur gleichen Klausel gehörenden Partner findet und weniger als quadratische Zeit (im Bezug auf die Formellänge) benötigt.

4. Aufgabe:
Beweisen Sie die folgenden Sachverhalte.

(a) Zu jeder Formel F gibt es eine äquivalente Formel G, die nur die Operatoren $\neg$ und $\rightarrow$ enthält.

(b) *Nicht* zu jeder Formel F gibt es eine äquivalente Formel G, die nur die Operatoren $\wedge$, $\vee$ und $\rightarrow$ enthält.