

Theorie der Programmiersprachen

3. Übung

1. Aufgabe:

Eine Formel sei als Text über dem Alphabet $\Sigma \cup \{(\, , \vee, \wedge, \neg\}$ gegeben, wobei die Variablen beliebige Teiltexthe über Σ sind.

Erklären Sie, wie man die Formel effizient in Adjazenzlistendarstellung mit Variablen A_1, \dots, A_n bringen kann.

2. Aufgabe:

Wir haben in der letzten Übung gesehen, dass der Polynomialzeitalgorithmus für 2-SAT aus der Vorlesung für eine Formel mit k Klauseln und n Variablen maximal $\mathcal{O}(n^2k)$ Zeit braucht.

In der Vorlesung haben wir gesehen, dass wir die Laufzeit mit Hilfe von Adjazenzlisten ohne andere Optimierungen auf $\mathcal{O}(n^3)$ beschränken können.

Zeigen Sie, dass wir die Laufzeit mit Adjazenzlisten ohne andere Optimierungen sogar auf $\mathcal{O}(nk)$ beschränken können.

3. Aufgabe:

Die Formel

$$F = (\neg A_1 \vee A_2) \wedge (A_2 \vee A_3) \wedge (\neg A_3 \vee A_4) \wedge (\neg A_4 \vee A_5) \wedge (\neg A_4 \vee \neg A_5) \wedge (A_3 \vee A_6) \wedge (A_3 \vee A_7)$$

ist gegeben. Bestimmen Sie mit Hilfe des Linearzeitalgorithmus für 2-KNF, ob die Formel erfüllbar ist.