

Theorie der Programmiersprachen

2. Übung

1. Aufgabe: Demonstrieren Sie den Lauf des Polynomialzeitalgorithmus für Hornformeln anhand eines selbstgewählten Beispiels. Geben Sie eine möglichst gute Laufzeitabschätzung bei Formelgröße m und Variablenzahl n an.

2. Aufgabe: (*Craig'scher Interpolationssatz*)

Es gelte $\models (F \rightarrow G)$ und es gibt mindestens eine atomare Formel, die sowohl in F als auch in G vorkommt. Man beweise, daß es eine Formel H gibt, die nur aus atomaren Formeln aufgebaut ist, die sowohl in F als auch in G vorkommen, mit $\models (F \rightarrow H)$ und $\models (H \rightarrow G)$.

Hinweis: Induktion über die Anzahl der atomaren Formeln, die in F , aber nicht in G vorkommen.

Andere Möglichkeit: Konstruieren einer Wahrheitstafel für H anhand der Wahrheitstafeln von F und G .

3. Aufgabe: Übersetzen Sie folgende aussagenlogische Formel in eine erfüllbarkeitsäquivalente 3-KNF.

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

4. Aufgabe: Sei L eine beliebige unendliche Menge von natürlichen Zahlen, dargestellt als Binärzahlen. Beweisen Sie, daß es eine unendliche Folge w_1, w_2, w_3, \dots von paarweise verschiedenen Binärzahlen gibt, so daß w_i Anfangsstück von w_{i+1} und von mindestens einem Element aus L ist ($i = 1, 2, 3, \dots$).

5. Aufgabe: Man gebe eine Formel an, zu der es keine äquivalente Hornformel gibt und begründe, warum dies so ist.

6. Aufgabe: Eine Formelmenge M_0 heißt ein *Axiomensystem* für eine Formelmenge M , falls

$$\{\mathcal{A} \mid \mathcal{A} \text{ ist Modell für } M_0\} = \{\mathcal{A} \mid \mathcal{A} \text{ ist Modell für } M\}.$$

M heißt *endlich axiomatisierbar*, falls es ein endliches Axiomensystem für M gibt. Es sei $\{F_1, F_2, F_3, \dots\}$ ein Axiomensystem für eine gewisse Menge M , wobei für $n = 1, 2, 3, \dots$ gilt: $\models (F_{n+1} \rightarrow F_n)$ und $\not\models (F_n \rightarrow F_{n+1})$. Man zeige: M ist nicht endlich axiomatisierbar. Zeigen Sie dafür zunächst: Die Menge $\{F_1, F_2, F_3, \dots\}$ enthält unendlich viele Atomformeln.