

Theorie der Programmiersprachen

2. Übung

1. 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 .

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

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

3. Aufgabe: Demonstrieren Sie den Lauf des Polynomialzeitalgorithmus für Hornformeln anhand der folgenden Formeln.

$$\begin{aligned} F &= (A \rightarrow 0) \wedge ((B \wedge C) \rightarrow A) \wedge (1 \rightarrow B) \wedge (B \rightarrow C) \\ G &= (A \rightarrow 0) \wedge ((B \wedge C) \rightarrow A) \wedge (1 \rightarrow C) \end{aligned}$$

Geben Sie eine möglichst gute Laufzeitabschätzung bei Formelgröße m und Variablenzahl n an.

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

5. Aufgabe: Beweisen Sie die folgenden Sachverhalte.

- Zu jeder Formel F gibt es eine äquivalente Formel G , die nur die Operatoren \neg und \rightarrow enthält.
- Nicht zu jeder Formel F gibt es eine äquivalente Formel G , die nur die Operatoren \wedge , \vee und \rightarrow enthält.