

Aufgaben zur Übergangsprüfung Grundlagen der Programmierung im WS 02/03
Zeit: 60 Minuten, erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den freien Stellen nach den jeweiligen Aufgaben ein (ggf. auf der jeweiligen Rückseite weiterschreiben). Sollten Unklarheiten oder Mehrdeutigkeiten bei der Aufgabenstellung auftreten, so notieren Sie bitte, wie Sie die Aufgabe interpretiert haben.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 5 Seiten.

Aufgabe 1.

Gegeben sei folgende aussagenlogische Formel: $((\neg A \vee \neg B) \wedge \neg (A \vee B))$.

- a) Transformieren Sie die Formel in eine äquivalente Formel in Klauselform.
- b) Prüfen Sie, ob die Formel erfüllbar ist.
- c) Ist die Formel auch tautologisch? Bitte begründen Sie Ihre Antwort.

Aufgabe 2.

- a) Geben Sie die Spezifikation für eine Funktion an, die als Eingabe ein Array der Länge N mit natürlichen Zahlen bekommt und bestimmt, ob in dem Eingabearray keine doppelten Elemente vorkommen.
- b) Geben Sie ein While-Programm zur Implementierung der unter a) spezifizierten Funktion an.
- c) Schätzen Sie die asymptotische Komplexität Ihres unter b) angegebenen Programmes ab.

Aufgabe 3.

Es seien Säugetier, Fleischfresser, Pflanzenfresser, Mensch, Pflanze und Vegetarier einstellige Prädikatensymbole. Weiterhin sei x eine Variable und tom eine Konstante.

Mit diesen syntaktischen Einheiten läßt sich eine Formelmenge F bilden:

$$F := \{ \begin{aligned} & \exists x (\text{Fleischfresser}(x) \wedge \text{Säugetier}(x) \wedge \text{Pflanze}(x)), \\ & \exists x (\text{Pflanzenfresser}(x) \wedge \text{Säugetier}(x)), \\ & \exists x (\text{Mensch}(x) \wedge \\ & \quad (\text{Fleischfresser}(x) \wedge \text{Pflanzenfresser}(x))), \\ & \exists x (\text{Vegetarier}(x) \wedge \\ & \quad (\text{Mensch}(x) \wedge \neg \text{Fleischfresser}(x))), \\ & \text{Vegetarier}(tom) \} \end{aligned}$$

- a) Argumentieren Sie, daß die Formel $\text{Säugetier}(tom)$ eine Folgerung aus F ist.
- b) Geben Sie ein beliebiges Modell für die Formelmenge F an.

Aufgabe 4.

Leider passierte es dem Professor Schusselig immer wieder, daß er vergißt, welche Aufgaben er den Studierenden gegeben hat. Leider sind die Lösungen auch nicht immer leicht zu verstehen. Nun hält er folgende Lösung des Studierenden Schlau in der Hand:

```
finde-index(a : array [1..N] of N0; i, j: N0): N0
  var m1, m2 : N0;
  if i >= j
  then i
  else m1, m2 := finde-index(a, i, i+((j-i)div 2)), finde-index(a, i+((j-i)div 2)+1, j);
    if a[m1] > a[m2]
    then m1
    else m2
    end if
  end if
```

```
f(a : array [1..N] of N0): N0
  a[finde-index(a, 1, N)]
```

- Um welche Art der Rekursion handelt es sich bei der Definition der Funktion `finde-index`?
- Nehmen wir an, N sei 3. Weiterhin sei das Array $b : \text{array } [1..N] \text{ of } N_0$ wie folgt initialisiert: $b := (100, 42, 101)$. Geben Sie die einzelnen Schritte zur Auswertung des Ausdrucks $f(b)$ an.
- Analysieren Sie Ihre Auswertung und helfen Sie dem Professor Schusselig. Welche Aufgabe hatte er gestellt, d.h. was sollte die Funktion f berechnen?
- Ist der Algorithmus des Studierenden Schlau in Hinblick auf die asymptotische Komplexität des Problems, das er lösen soll, optimal? Bitte begründen Sie Ihre Antwort.