Fachhochschule Wedel
Prof. Dr. Sebastian Iwanowski

21.08.2018

FLV

Aufgaben zur Klausur in

Grundlagen der Theoretischen Informatik,

***Teil Formale Logik und Verifikation* (SS 2018)**

Studiengänge B_Inf14.0, B_WInf14.0

Zeit: 60 Minuten (für diesen Teil), 120 Minuten für die gesamte Klausur

erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den freien Stellen nach den jeweiligen Aufgaben ein (ggf. auf der gegenüberliegenden Rückseite weiterschreiben).

Diese Klausur besteht einschließlich dieses Deckblatts aus 5 Seiten.

Für die Klausur werden insgesamt 30 Bewertungseinheiten (BE) vergeben. Ihr prozentuales Ergebnis in dieser Klausur wird 1:1 mit dem prozentualen Ergebnis in Automaten und Formale Sprachen verrechnet. Sie müssen zum Bestehen insgesamt 50% erzielen.

Viel Erfolg !

1. Aufgabe (3 BE)

Gegeben sei die Formel F: $q \vee \neg (q \rightarrow r)$

Bringen Sie F in eine konjunktive Normalform mit möglichst wenigen Klauseln.

2. Aufgabe (8 BE)

Gegeben seien die folgenden Prädikate mit den zugehörigen Bedeutungen:

L(x,y) x liebt y

S(x,n) x ist im n-ten Semester

B(x,v) x besteht Vorlesung v (Gegenteil: fällt durch)

H(x,y,v) x hilft y in Vorlesung v

Drücken Sie folgende Sachverhalte durch eine Verknüpfung der oben stehenden Prädikate aus:

a) Anna ist eine Studentin im 1. Semester. (1 BE)

b) Anna lässt sich von Klaus in Vorlesung GTI helfen, aber sie wird von Ernst geliebt, doch der hilft nicht, denn der ist in GTI durchgefallen. (2 BE)

c) Alle Studierenden, die sich in irgendeiner Vorlesung von jemandem helfen lassen, der selbst darin bestanden hat, bestehen diese Vorlesung auch. (2,5 BE)

d) Wenn ein Studierender sich von jemandem in einer Vorlesung helfen lässt und diese Vorlesung besteht, dann liebt der Studierende seinen Helfer. (2,5 BE)

3. Aufgabe (6 BE)

a) Finden Sie zum folgenden Programmausschnitt und der gegebenen Nachbedingung die schwächste Vorbedingung. Vereinfachen Sie diese so weit wie möglich. Geben Sie alle Zwischenschritte Ihrer Beweiskette an (5 BE)

b) Geben Sie eine Belegung für x und y an, welche die gegebene Nachbedingung erfüllt und die then-Anweisung durchläuft und geben Sie eine Belegung an, welche die gegebene Nachbedingung erfüllt und die else-Anweisung durchläuft! (1 BE)

```
if y = x

   then

      y := x * y

   else

      y := x - y;
```

N ⇔ y = x

4. Aufgabe (7 BE)

Betrachten Sie folgendes Programm:

```
{ s,t ∈ N }
s := 0;
t := 1;
while(t < 10) do
   begin
      s := 2 • t • s;
      t := 2 • (t + s);
  end
```

a) Formulieren Sie Bedingungen für s und t, die für jeden Schleifendurchlauf gültig sind. Beweisen Sie das durch vollständige Induktion. (5 BE)

b) Zeigen Sie, dass dann die Schleife irgendwann terminiert (wann genau?). Formulieren und beweisen Sie, was diese dann berechnet hat. (2 BE)

5. Aufgabe (6 BE)

Gegeben sei die folgende Funktion:

```
procedure f (n, m: integer): integer
begin
  if (m ≤ 0)
    then return n
    else return f (n + 1, m - 1)
end {f}
```

a) Von welchem Typ ist die Rekursion? Geben Sie eine Begründung an! (1 BE)

b) Was berechnet diese Funktion in Abhängigkeit von n und m? Beweisen Sie Ihre Behauptung durch vollständige Induktion! (4 BE)

c) Wandeln Sie die Implementierung in eine äquivalente nichtrekursive um!
Anmerkung: Hier ist nicht nur nach einer Implementierung gefragt, die dasselbe ausgibt, sondern die denselben Algorithmus realisiert. (1 BE)