Aufgaben zur Klausur in

Grundlagen der Theoretischen Informatik,

***Teil Formale Logik und Verifikation* (SS 2016)**

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 6 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 (4 BE)

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

a) Bringen Sie F in eine konjunktive Normalform mit möglichst wenigen Klauseln und stellen Sie diese in Mengendarstellung dar! (3 BE)

b) Von welchem Typ ist F?

Zur Auswahl stehen die 3 Kategorien Tautologie, erfüllbar und widerlegbar, Widerspruch. Begründen Sie Ihre Antwort! (1 BE)

2. Aufgabe (9 BE)

Gegeben seien die folgenden Prädikate auf der Menge aller Menschen:

V (x,y): x ist Vater von y

M (x,y): x ist Mutter von y

S(x): x studiert oder hat studiert

A(x): x hat eine Arbeit

a) Drücken Sie die folgenden Sachverhalte ausschließlich durch eine prädikatenlogische Verknüpfung dieser vier Prädikate aus!

i) Hugo hat studiert, Anna dagegen nicht. (1 BE)

ii) Kinder von Eltern, die beide studiert haben, studieren ebenfalls. (2 BE)

iii) Annas Kind Lisa studiert. (2 BE)

Zur Info: Anna ist ein weiblicher Vorname.

iv) Nicht jeder, der studiert, bekommt eine Arbeit. (2 BE)

b) Geben Sie in einem allgemeinverständlichen Satz an, was folgende Formel bedeutet:

$\forall x \, \forall y \, \exists z \, (V(x, z) \wedge M(y,z) \wedge S(x) \wedge S(y) \wedge S(z))$ (2 BE)

3. Aufgabe (5 BE)

a) Finden Sie zum folgenden Programmausschnitt und der gegebenen Nachbedingung die schwächste Vorbedingung!
Vereinfachen Sie die Bedingungen so weit wie möglich (aber nicht noch weiter!)
Geben Sie alle Zwischenschritte Ihrer Beweiskette an! (4 BE)

```
if x > y
   then
      begin


         y := x • y;

      end
   else
      begin


         x := x • y;

      end
```

$x = y^2$

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)

4. Aufgabe (8 BE)

Gegeben sei die folgende Funktion f:

```
procedure f (x: integer): integer
begin
  m := 0;
  n := x;
  while (n > 0) do
    begin
      m := m + n;
      n := n - 1;
    end;
  return n;
end {f}
```

a) Seien n_i und m_i die Werte der Variablen nach dem i-ten Schleifendurchlauf. Bestimmen Sie diese Werte durch eine nichtrekursive Formel (abhängig von i und eventuell dem Parameter x) und beweisen Sie diese durch vollständige Induktion über i.

(6 BE)

b) Folgern Sie aus a), wann die Schleife abbricht und folgern Sie daraus, welchen Wert f letztendlich in Abhängigkeit von x zurückgibt. Geben Sie gegebenenfalls eine Vorbedingung an, die für Ihre Aussage gelten muss! (2 BE)

5. Aufgabe (4 BE)

Bestimmen Sie die Rekursionstypen der folgenden Funktionen und begründen Sie Ihre Antwort (geben Sie den jeweils speziellsten Typ an, der möglich ist):

a)

```
procedure fib (a: ℕ, b: ℕ, n: ℕ): ℕ
begin
  if (n = 0)
    then return b
    else return fib (a+b, a, n-1)
end {f}
```

b)

```
procedure f (x, y: ℕ): ℕ
begin
  if (x = 0)
    then return 1
    else return x * f (x, y-1)
end {f}
```