

---

**Aufgaben zur Klausur in**  
***Grundlagen der Theoretischen Informatik (SS 2009)***  
**Studiengänge B\_Inf, B\_TInf, B\_MInf, B\_WInf**

Zeit: 90 Minuten,  
erlaubte Hilfsmittel: keine

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

Diese Klausur besteht einschließlich dieses Deckblatts aus 6 Seiten.

Für die Klausur werden insgesamt 50 Bewertungseinheiten (BE) vergeben. Zum Bestehen benötigen Sie mindestens 25 BE.

Viel Erfolg !

## 1. Aufgabe (12 BE)

- a) Bringen Sie die folgenden Formeln in konjunktive Normalform! Vereinfachen Sie den Ausdruck so weit wie möglich! Geben Sie für Ihre Vereinfachung an, wie viele Klauseln die Formel enthält!
- b) Geben Sie jeweils an, welcher der Sachverhalte i) – iii) auf die Formel zutrifft (genau einer) und begründen Sie Ihre Antwort genau:
- i) Formel ist gültig (Tautologie)
  - ii) Formel ist erfüllbar, aber keine Tautologie
  - iii) Formel ist widersprüchlich

1.1  $(p \vee \neg q) \vee (z \wedge \neg p)$

1.2  $(p \wedge \neg q) \vee (q \wedge \neg p)$

1.3  $(p \wedge \neg q) \rightarrow (q \vee p)$

## 2. Aufgabe (10 BE)

Gegeben seien die folgenden Prädikate mit den zugehörigen Bedeutungen:

$L(x,y)$        $x$  liebt  $y$

$F(x)$            $x$  ist weiblich (= Frau)

$M(x)$            $x$  ist männlich (= Mann)

Hierbei sei der Definitionsbereich für alle Eingaben die Menge aller Menschen.

Drücken Sie folgende Sachverhalte durch eine Verknüpfung der oben stehenden Prädikate aus (Sie dürfen nicht voraussetzen, dass die Prädikate symmetrisch sind oder dass sich Mann und Frau ausschließen oder sonst irgendetwas, das hier nicht explizit definiert ist):

2.1 Jede Frau wird von (irgend)einem Mann geliebt.

2.2 Einige Frauen lieben Hans.

2.3 Nur Frauen lieben Hans.

2.4 Kein Mann liebt Hans.

2.5 Jeder Mann liebt sich selbst.

### 3. Aufgabe (8 BE)

- a) Finden Sie zum folgenden Programmausschnitt und der gegebenen Nachbedingung die schwächste Vorbedingung! Geben Sie alle Zwischenschritte Ihrer Beweiskette an! (6 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! (2 BE)

```
if y < x
```

```
  then
```

```
    y := x * y
```

```
  else
```

```
    y := x + y;
```

```
{y < x}
```

## 4. Aufgabe (8 BE)

Gegeben sei der folgende Programmausschnitt:

```
{n > 0 vom Typ integer}
s := 1; k := n;
while (k > 0) do
begin
    k := k - 1;
    s := k + s
end
```

- a) Was berechnet dieser Programmausschnitt: Geben Sie eine Nachbedingung für  $s$  zur gegebenen Vorbedingung an! (2 BE)
- b) Verifizieren Sie Ihre Aussage von a) nach folgendem Verfahren: Formulieren Sie Bedingungen, die nach dem  $i$ -ten Schleifendurchlauf erfüllt sind. Beweisen Sie diese Bedingungen mit vollständiger Induktion über  $i$  und zeigen Sie, dass aus den Bedingungen die Aussage von a) folgt. (6 BE)

## 5. Aufgabe (12 BE)

Gegeben sei die folgende Funktion:

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

a) Von welchem Typ ist die Rekursion? Geben Sie eine Begründung an! (2 BE)

b) Nennen Sie eine Vorbedingung, die erfüllt sein muss, damit f überhaupt einen Wert berechnet! Geben Sie an, wie sich die Implementierung verhält, wenn diese Vorbedingung nicht erfüllt ist! (2 BE)

c) Was berechnet diese Funktion, wenn die Vorbedingung erfüllt ist, in Abhängigkeit von n und m? Beweisen Sie Ihre Behauptung durch vollständige Induktion! Hinweis: Geben Sie die Größe an (es ist nur eine!), nach der Sie die Induktion führen! (5 BE)

d) 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. (3 BE)