
Aufgaben zur Klausur in
Grundlagen der Theoretischen Informatik (SS 2011)
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 (8 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 $(p \vee \neg q) \vee (q \wedge \neg p)$ (2 BE)

1.2 $(q \wedge \neg q) \vee (\neg q \rightarrow q)$ (3 BE)

1.3 $p \rightarrow (\neg q \rightarrow z)$ (3 BE)

2. Aufgabe (12 BE)

Gegeben seien die folgenden Prädikate mit den zugehörigen Bedeutungen:

$L(x,y)$ x liebt y

$F(x)$ x ist weiblich

$S(x,n)$ 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

- a) Schreiben Sie neben alle Prädikate sinnvolle Definitionsbereiche (1 BE)
- b) Drücken Sie folgende Sachverhalte durch eine Verknüpfung der oben stehenden Prädikate aus:

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

2.2 Anna lässt sich von Klaus in Vorlesung GTI helfen, aber sie liebt Ernst, doch der ist in GTI durchgefallen. (2 BE)

2.3 Alle Studierenden, die sich in irgendeiner Vorlesung von jemandem helfen lassen, der selbst darin bestanden hat, bestehen diese Vorlesung auch. (3 BE)

2.4 Wenn ein Studierender sich von jemandem in einer Vorlesung helfen lässt und diese Vorlesung besteht, dann liebt der Studierende seinen Helfer. (3 BE)

2.5 Klaus liebt eine Studentin im 3. Semester (wir wollen nicht verraten wen). (2 BE)

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 (13 BE)

Gegeben sei der folgende Programmausschnitt:

```
{n > 0 vom Typ integer}
```

```
s := n; k := n;
```

```
while (k > 1) do
```

```
begin
```

```
    s := s / k;
```

```
    k := k - 1;
```

```
    s := k * s;
```

```
    s := k * s
```

```
end
```

a) Formulieren Sie Invariantenbedingungen, die nach jedem Schleifendurchlauf erfüllt sind in Abhängigkeit von der Anzahl der durchgeführten Schleifendurchläufe (bezeichnet durch i). (3 BE)

b) Beweisen Sie Ihre Aussage von a) mit vollständiger Induktion. (7 BE)

c) Geben Sie an, nach wie vielen Durchläufen die Schleife abbricht und folgern Sie daraus die endgültigen Werte für alle Variablen dieses Programms. (2 BE)

d) Was berechnet das Programm, wenn Sie die Bedingung $(k > 1)$ durch $(k \geq 1)$ ersetzen? (1 BE)

5. Aufgabe (9 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) 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! (4 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. (2 BE)