



Aufgabenblatt 1

Abgabe: 29.04.06, 24.00 Uhr

- Wie aus dem ersten Semester gewohnt, finden Sie alle wichtigen Informationen und die Übungsblätter auf der Vorlesungsseite:
<http://www.informatik.uni-stuttgart.de/fmi/fk/lehre/ss06/info2/default.htm>
Im eClaus-System sind ihre Accounts erhalten geblieben, Sie müssen sich für die Übungen daher keinen neuen Account anlegen! Nachdem Sie sich wie gewohnt eingeloggt haben, müssen Sie sich stattdessen unter:
Allgemeine Funktionen → Meine persönlichen Daten
für die Veranstaltung *Einführung in die Informatik II* anmelden (das Passwort lautet: `claus_ss06`).
- Zusätzlich gibt es dieses Semester einen Multiple-Choice eClaus (MC eClaus)! Dabei sollen Sie jede Vorlesungswoche MC-Aufgaben alleine beantworten, sie dienen als zusätzliche Lernkontrolle. Den Link dazu finden Sie auf der Vorlesungsseite. Aus technischen Gründen müssen Sie sich für das MC eClaus einen separaten Account anlegen, das Passwort lautet: `baum05`.
- Auch im zweiten Semester wird in den Übungen das kooperative Lernkonzept angewendet. Der organisatorische Ablauf ist:
 - 1a) Sie melden sich für die Übungen zur Vorlesung im eClaus-System an (siehe oben).
 - 1b) Sie legen sich ein Account im MC eClaus System an (siehe oben).
 - 2a) In der ersten Vorlesungswoche finden noch keine Übungen statt!
 - 2b) In der ersten Vorlesungswoche bearbeiten Sie dieses erste Aufgabenblatt (noch) alleine und geben Sie bis Samstag 24 Uhr online ab.
 - 2c) Sie können mit dem ersten Aufgabenblatt des MC eClaus beginnen.
 - 3) Am Sonntag Abend erhalten Sie von Ihrem Tutor eine E-Mail, aus der hervorgeht, welcher Lerngruppe Sie angehören (drei bis vier Studierende pro Lerngruppe).
 - 4) Ab der zweiten Vorlesungswoche geht alles seinen gewohnten Gang: Aufgabenblatt 2 bearbeiten, Lerngruppe treffen und in die Übung gehen (Aufgabenblatt 1 wird besprochen).
- Ein Übungsschein wird ausgestellt, sofern
 - mindestens 60 % (228 Punkte) der Gesamtpunktzahl erreicht wurden (Gesamtpunktzahl 380 = 240 (12 Aufgabenblätter) + 60 (2 Testklausuren + 80 MC eClaus)),
 - in den letzten vier Übungsblättern die Hälfte der Punkte erreicht wurden,
 - der/die Betreffende höchstens zwei Mal unentschuldigt bei den Übungen gefehlt hat,
 - der/die Betreffende mindestens drei Aufgaben korrekt an der Tafel vorgebracht hat,

- der/die Betreffende mindestens die Hälfte der Punkte in den Testklausuren erreicht hat!

Ein Gesamtübungsschein für Studierende der Studiengänge Informatik und Softwaretechnik wird nur ausgestellt, wenn der/die Betreffende auch den ersten Teilübungsschein erhalten hat.

1. Deterministische k-Band-Turingmaschine (mittel) (schriftlich, 5 Punkte)

Sie haben gegen Ende des ersten Semester Berechnungsmodelle kennengelernt. Im Kapitel 6.2 wird die zeichenorientiert arbeitende Turingmaschine ausführlich vorgestellt.



Alan Turing
(1912-1954)

Konstruieren Sie eine deterministische Turingmaschine mit dem Eingabealphabet $\Sigma = \{a, b\}$, die in jedem Eingabewort w je zwei aufeinander folgende a durch ein b ersetzt. Das Eingabewort $w = aab$ muss also in bb umgewandelt werden, das Eingabewort $w = abaabbaaab$ in $abbbbab$ und $w = aaaa$ in bb .

Geben Sie die Übergangsfunktion δ in Tabellenform im eClaus System ab.

2. Axiomatische Semantik I (leicht-mittel) (schriftlich, 5 Punkte)

Kreuzen Sie jeweils die richtigen Aussagen an, alle Variablen sind vom Typ **integer**. Geben Sie die Lösung im eClaus System in folgendem Format ab: 'a) 1100' (für die Teilaufgabe a) z.B.).

- a) Zur Vorbedingung und dem Programmfragment
 $\{x-y < 0\} \quad x := y + 4$
sind die folgenden Zusicherungen gültige Nachbedingungen:
 $\{x < y\}$ $\{x > 0 \text{ and } y > 0\}$ $\{x = y + 4\}$ $\{x \geq y\}$
- b) Zur Vorbedingung und dem Programmfragment
 $\{x > 42\} \quad x := x - 4$
sind die folgenden Zusicherungen gültige Nachbedingungen:
 $\{x > 0\}$ $\{x > 46\}$ $\{x > 38\}$ $\{x \neq 42\}$
- c) Zur Vorbedingung und dem Programmfragment
 $\{x \bmod 2 = 0\} \text{ if } x > 6 \text{ then } x = x * 3; \text{ end if};$
sind die folgenden Zusicherungen gültige Nachbedingungen:
 $\{x \bmod 2 = 0\}$ $\{x \bmod 6 = 0\}$ $\{x > 18\}$ $\{x > 2\}$
- d) Zur Vorbedingung und dem Programmfragment
 $\{x < 7\} \text{ if } x > -4 \text{ then } x := -x; \text{ else } x := x + 7; \text{ end if};$
sind die folgenden Zusicherungen gültige Nachbedingungen:
 $\{x > -7\}$ $\{x > 0\}$ $\{x \leq 3\}$ $\{x < 7\}$
- e) Im folgenden Programmfragment
 $\text{while } x > 42 \text{ loop } x := x - 42; \text{ end loop};$
sind die folgenden Zusicherungen gültige Invarianten:
 $\{h = 12\}$ $\{1 + 1 = 2\}$ $\{x > 0\}$ $\{x > 42\}$

3. Axiomatische Semantik II (leicht-mittel) (schriftlich, 5 Punkte)
 Kreuzen Sie jeweils die richtigen Aussagen an, alle Variablen sind vom Typ **integer**.
 Geben Sie die Lösung im eClaus System in folgendem Format ab: 'a) 1100' (für die Teilaufgabe a) z.B.).

- a) Zur Nachbedingung und dem Programmfragment
 $x:=2*y+6 \{x>42\}$
 sind die folgenden Zusicherungen schwächste Vorbedingungen:
 $\{y>18\}$ $\{y>24\}$ $\{x>96\}$ $\{y>96\}$
- b) Zur Nachbedingung und dem Programmfragment
 $x:=x-3 \{x>42\}$
 sind die folgenden Zusicherungen schwächste Vorbedingungen:
 $\{x>0\}$ $\{x>45\}$ $\{x>39\}$ $\{x=x-3\}$
- c) Zur Nachbedingung und dem Programmfragment
 $\text{if } x>6 \text{ then } x:=6; \text{ else } x:=3; \text{ end if}; \{x=3\}$
 sind die folgenden Zusicherungen schwächste Vorbedingungen:
 $\{x=3 \text{ or } x=6\}$ $\{x>6\}$ $\{x\leq 6\}$ $\{x=3 \text{ and } x=6\}$
- d) Zur Nachbedingung und dem Programmfragment
 $\text{if } x>4 \text{ then } y:=x-12; \text{ else } y:=4+x; \text{ end if}; \{y>12\}$
 sind die folgenden Zusicherungen schwächste Vorbedingungen:
 $\{y>12\}$ $\{x>4\}$ $\{x>8\}$ $\{x>24\}$
- e) Zur Nachbedingung und dem Programmfragment
 $\text{while } x>42 \text{ loop } x:=x-42; \text{ end loop } \{x>10\}$
 sind die folgenden Zusicherungen schwächste Vorbedingungen:
 $\{x>42\}$ $\{x>52\}$ $\{x>10 \text{ and } x \bmod 42 > 10\}$ $\{x>10\}$

4. Axiomatische Semantik III (mittel) (schriftlich, 3+2+1 Punkte)
 In vielen Abschätzungen zur Laufzeit von Algorithmen taucht die Summe der natürlichen Zahlen bis zu einer Zahl n oder $n - 1$ auf. Betrachten Sie das folgende Programm (alle Variablen sind als **integer** deklariert):

```
s:=0; Get(n);
while n>0 loop
  s:=s+n;
  n:=n-1;
end loop;
Put(s);
```

- a) Welche Ausgabe liefert das Programm in Abhängigkeit von der eingelesenen Zahl n ? (Kein Summenzeichen verwenden! – ausrechnen!) Geben Sie eine Invariante an, mit deren Hilfe und den Hoareschen Regeln sich die Korrektheit Ihrer Vermutung zeigen lässt.
- b) Führen Sie den Beweis Ihrer Vermutung aus der ersten Teilaufgabe mit Hilfe Ihrer Invariante und den Hoareschen Regeln durch.
- c) Wie verändert sich die berechnete Funktion und der Korrektheitsbeweis, wenn die beiden Anweisungen des Schleifenrumpfes vertauscht werden?