

Aufgabe 1 (H) (Berechnung von Verifikationsbedingungen)

Berechnen Sie zu folgenden Programmen und Nachbedingungen jeweils die Vorbedingung (pre) und die Verifikationsbedingung (vc).

a) Programm: **if** $X > 0$ **then** $Y := X$ **else** $Y := -X$
Nachbedingung: $\{Y \leq 5\}$

b) Programm:

$N := 0; M := 0; K := 0;$
while $K < a$ **do** $\{I\} (N := N + 1; K := K + M + 1; M := M + 2)$

Nachbedingung: $\{N - 1 \leq \sqrt{a} \leq N\}$

Führen Sie die Berechnung für

- i) die richtige Invariante,
- ii) eine zu schwache Invariante und
- iii) eine zu starke Invariante

durch.

Aufgabe 2 (H) (Totale Korrektheit)

Gegeben sei folgendes Programm:

$\{0 \leq n_0\}$
 $N := 0; P := 1; C := 0;$
while $\neg(N = n_0)$ **do**
 if $C = 0$ **then** $(N := N + 1; C := P)$
 else $(C := C - 1; P := P + 1);$
 $P := P + C$
 $\{P = ?\}$

- a) Was ist der Wert von P nach Ausführung des Programms?
- b) Wie lautet die Invariante der Schleife?
- c) Zeigen Sie die totale Korrektheit des Programms. Benutzen Sie dazu folgende Regel für die While-Schleife:

$$\frac{\{I \wedge b \wedge t = z\} s \{I \wedge t < z\}}{\{I\} \mathbf{while} b \mathbf{do} s \{I \wedge \neg b\}}$$

Der Term t darf beliebige aus der Mathematik bekannte Funktionen enthalten.

Aufgabe 3 (Ü) (Normalisierung von Hoare-Beweisen)

Die Regeln des Hoare-Kalküls sind bis auf die Konsequenzregel syntaxgesteuert (es existiert jeweils genau eine Regel für jedes Statement). Obwohl die Konsequenzregel prinzipiell immer anwendbar ist, kann man sich in der Beweissuche jedoch auf bestimmte Situationen beschränken.

Wir betrachten dazu folgendes System K aus der Semikolon-Regel, der **if**-Regel und den abgeleiteten Regeln

$$\frac{P \Longrightarrow Q}{\{P\} \text{ skip } \{Q\}} \quad \frac{P \Longrightarrow Q[a/x]}{\{P\} x := a \{Q\}}$$
$$\frac{P \Longrightarrow I \quad \{I \wedge b\} s \{I\} \quad I \wedge \neg b \Longrightarrow Q}{\{P\} \text{ while } b \text{ do } s \{Q\}}$$

Zeigen Sie nun folgende Behauptung: Jeder Beweis im Hoare-Kalkül kann in einen Beweis im System K überführt werden. Formal: $\vdash \{P\}s\{Q\} \Longrightarrow \vdash_K \{P\}s\{Q\}$

Aufgabe 4 (P) (Generator für Verifikationsbedingungen)

Implementieren Sie den Generator für Verifikationsbedingungen aus der Vorlesung. Es bietet sich die Implementierung in einer funktionalen Programmiersprache an. Einen Rahmen für das Programm in Gofer finden Sie in folgender Datei.

```
/usr/proj/semantik/gofer/vc-gen/rahmen.gs
```

Sie können die Lösung zu der Aufgabe aber auch in Prolog abgeben.

Prüfung

Wie bereits angekündigt findet die studienbegleitende Prüfung am Mittwoch, den 18. Juli statt. Anmeldung bitte per E-Mail bis spätestens Donnerstag, den 12. Juli an ballarin@in.tum.de unter Angabe von Name, Matrikelnummer, Geburtsdatum und Studiengang. Sie erhalten dann eine Bestätigung mit Ihrem Prüfungstermin per E-Mail.