

Aufgabe 1 (H) (*Monotonie von pre und vc*)

Zeigen Sie, dass die in der Vorlesung definierten Funktionen *pre* und *vc* monoton bezüglich der Implikation sind, d.h. dass für alle Programme *s* und Zusicherungen *P* und *Q* gilt:

- a) $(P \implies Q) \implies (\text{pre } s P \implies \text{pre } s Q)$
- b) $(P \implies Q) \implies (\text{vc } s P \implies \text{vc } s Q)$

Hinweis: Strukturelle Induktion über *s*. Sie dürfen verwenden, dass die Substitution monoton bezüglich der Implikation ist, d.h. dass für alle arithmetischen Ausdrücke *a* und Programmvariablen *X* gilt: $(P \implies Q) \implies (P[a/X] \implies Q[a/X])$.

Aufgabe 2 (Ü) (*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 3 (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.