

PROGRAMMIERUNG

ÜBUNG 12: HOARE-KALKÜL

Eric Kunze

`eric.kunze@tu-dresden.de`

TU Dresden, 06. Juli 2022

letzte Änderung:
06.07.2022, 16:31

1. Funktionale Programmierung
 - 1.1 Einführung in Haskell: Listen
 - 1.2 Algebraische Datentypen
 - 1.3 Funktionen höherer Ordnung
 - 1.4 Typpolymorphie & Unifikation
 - 1.5 Beweis von Programmeigenschaften
 - 1.6 λ -Kalkül
2. Logikprogrammierung
3. Implementierung einer imperativen Programmiersprache
 - 3.1 Implementierung von C_0
 - 3.2 Implementierung von C_1
4. **Verifikation von Programmeigenschaften**
5. H_0 – ein einfacher Kern von Haskell

Hoare-Kalkül

- ▶ Beweis / Verifikation von Programmeigenschaften
- ▶ Verifikationsformeln der Form $\{P\} \mathbf{A} \{Q\}$
 - ▷ P und Q sind Zusicherungen (prädikatenlogische Ausdrücke)
 - ▷ P heißt **Vorbedingung**, Q heißt **Nachbedingung**
 - ▷ Beschreibung der Veränderung von Zusicherungen
 - ▷ **Bedeutung:** Wenn die Variablenwerte vor Ausführung von \mathbf{A} die Zusicherung P erfüllen und \mathbf{A} terminiert, dann erfüllen die Variablen nach Ausführung von \mathbf{A} die Zusicherung Q
- ▶ Aufstellen eines Beweisbaumes mit zur Verfügung stehenden Regeln

- ▶ Zuweisungsaxiom
- ▶ Sequenzregel
- ▶ CompRegel
- ▶ Iterationsregel
- ▶ (erste und zweite) Alternativregel
- ▶ Konsequenzregeln
 - ▷ stärkere Vorbedingung
 - ▷ schwächere Nachbedingung

SCHLEIFENINVARIANTE

Für die Iterationsregel benötigen wir die Schleifeninvariante SI . In den meisten unserer Fälle ist diese von der Form $SI = A \wedge B$, wobei

- ▶ A den Zusammenhang zwischen Zählvariable und Akkumulationsvariablen beschreibt. Führe dazu einige Iterationen der Schleife durch und leite daraus einen Zusammenhang her.
- ▶ B die abgeschwächte Schleifenbedingung ist. Dabei nehmen wir die letztmögliche Variablenbelegung, für die die Schleifenbedingung π noch wahr ist und führen den Schleifenrumpf noch einmal darauf aus ($\rightarrow \pi'$).

$$\rightsquigarrow B = \pi \cup \pi'$$

Aufgabe 1

AUFGABE 1 – TEIL (A)

Verifikationsformel:

$$\underbrace{\left\{ \begin{array}{l} (k \geq 0) \wedge (u \geq k) \\ \wedge (j = k) \wedge (s = 0) \end{array} \right\}}_{\text{Vorbedingung}} \text{ while } (j < u) \{ j=j+1; s=j+s;; \} \underbrace{\left\{ s = \frac{u^2 + u - k^2 - k}{2} \right\}}_{\text{Nachbedingung}}$$

Schleifeninvariante: $SI = A \wedge B$

#	j	s
0	k	0
1	k + 1	(k + 1)
2	k + 2	(k + 2) + (k + 1)
⋮	⋮	⋮
N	k + N	(k + N) + ⋯ + (k + 1)

Als Gleichungssystem:

$$\begin{aligned} j &= k + N \\ s &= \sum_{i=k+1}^{k+N} i \\ \Rightarrow A &= (s = \sum_{i=k+1}^j i) \end{aligned}$$

AUFGABE 1 – TEIL (A)

$SI = A \wedge B$ und wir wissen schon $A = (s = \sum_{i=k+1}^j i)$

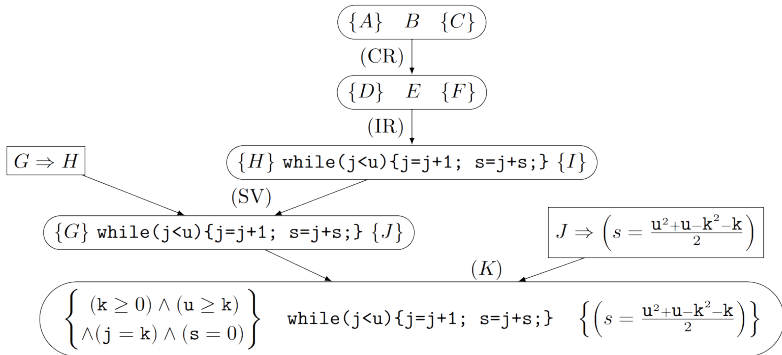
abgeschwächte Schleifenbedingung:

- ▶ Schleifenbedingung: $\pi = (j < u)$
- ▶ Schleifenbedingung letztmalig wahr für $j = u - 1$
- ▶ Wert nach nochmaligem Schleifendurchlauf:
 $\pi' = (j = u)$
- ▶ $B = \pi \cup \pi' = (j \leq u)$ *(symbolische Schreibweise)*

$$\implies SI = A \wedge B = (s = \sum_{i=k+1}^j i) \wedge (j \leq u)$$

AUFGABE 1 – TEIL (B)

Sei $SI = A \wedge B = \left(s = \sum_{i=k+1}^j i \right) \wedge (j \leq u)$ und $\pi = (j < u)$.



AUFGABE 1 – TEIL (B)

Verifikationsformel:

$$\left\{ \begin{array}{l} (k \geq 0) \wedge (u \geq k) \\ \wedge (j = k) \wedge (s = 0) \end{array} \right\} \text{ while } (j < u) \{ j=j+1; s=j+s;; \} \left\{ s = \frac{u^2 + u - k^2 - k}{2} \right\}$$

Sei $SI = A \wedge B = \left(s = \sum_{i=k+1}^j i \right) \wedge (j \leq u)$ und $\pi = (j < u)$.

$$A = D = SI \wedge \pi = SI \wedge (j < u)$$

$$B = j = j + 1; s = j + s$$

$$C = F = H = SI$$

$$E = \{ j = j + 1; s = j + s \}$$

$$G = (k \geq 0) \wedge (u \geq k) \wedge (j = k) \wedge (s = 0)$$

$$I = J = SI \wedge \neg \pi = SI \wedge \neg(j < u)$$

K = schwächere Nachbedingung (SN)

AUFGABE 2

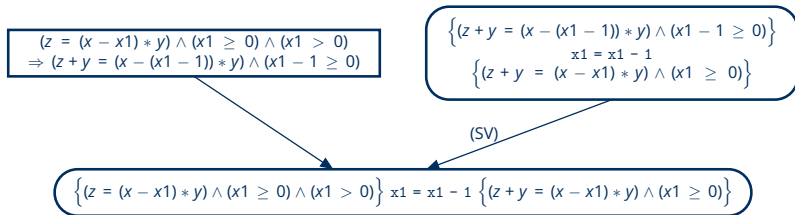
Zeigen Sie die Gültigkeit der Verifikationsformel

$$\left\{ (z = (x - x1) \cdot y) \wedge (x1 \geq 0) \wedge (x1 > 0) \right\}$$

$$x1 = x1 - 1;$$

$$\left\{ (z + y = (x - x1) \cdot y) \wedge (x1 \geq 0) \right\}.$$

AUFGABE 2

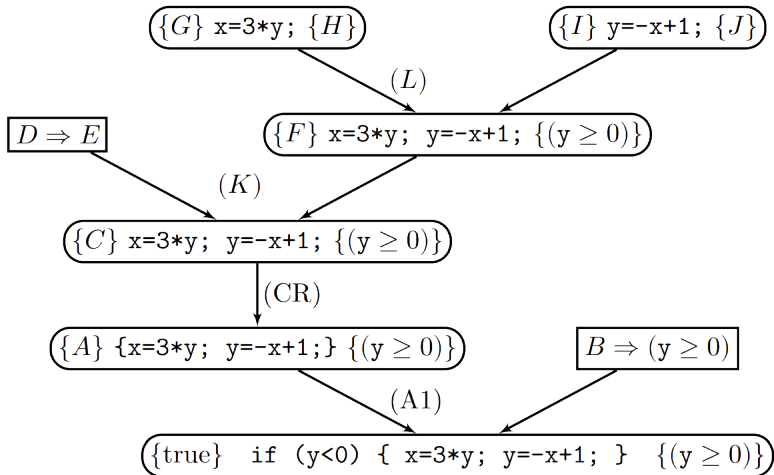


wobei (beachte: x_1 ist Ganzzahl)

$$\begin{aligned} & (z = (x - x_1) \cdot y) \wedge (x_1 \geq 0) \wedge (x_1 > 0) \\ \Rightarrow & (z + y = (x - x_1) \cdot y + y) \wedge (x_1 \geq 0) \wedge (x_1 > 0) \\ \Rightarrow & (z + y = (x - x_1 + 1) \cdot y) \wedge (x_1 \geq 0) \wedge (x_1 > 0) \\ \Rightarrow & (z + y = (x - (x_1 - 1)) \cdot y) \wedge (x_1 \geq 0) \wedge (x_1 > 0) \\ \Rightarrow & (z + y = (x - (x_1 - 1)) \cdot y) \wedge (x_1 \geq 0) \wedge (x_1 - 1 \geq 0) \end{aligned}$$

Aufgabe 3

AUFGABE 3 – TEIL (A)



AUFGABE 3 – TEIL (A)

$$A = \text{true} \wedge (y < 0)$$

$$B = \text{true} \wedge \neg (y < 0)$$

$$C = A$$

$$D = A$$

$$E = -(3 \cdot y) + 1 \geq 0$$

$$F = E$$

$$G = E$$

$$H = (-x + 1 \geq 0)$$

$$I = H$$

$$J = (y \geq 0)$$

$$K = \text{stärkere Vorbedingung}$$

$$L = \text{Sequenzregel}$$

AUFGABE 3 – TEIL (B)

zu zeigen: $\text{true} \wedge (y < 0) \Rightarrow (-3 \cdot y + 1 \geq 0)$

$$\text{true} \wedge (y < 0) \Rightarrow y < 0$$

$$\Rightarrow -3 \cdot y > 0$$

$$\Rightarrow -3 \cdot y + 1 > 1$$

$$\Rightarrow -3 \cdot y + 1 \geq 0$$