Technische Universität München **WS 2006/07**
Institut für Informatik
Prof. Manfred Broy
A. Gruler, B. Spanfelner **Musterlösung zu Blatt 10**

Übungen zu „Formale Methoden“

Aufgabe 1 Beweisverfahren über Schleifen [LÖSUNG]

(a) Gemäß Aufgabenstellung muss am Ende des Programms das Prädikat R gelten:

$$R \quad \equiv \quad \Big(s = \sum\nolimits_{k:\ 0 \le k < 11} b[k]\Big)$$

Das Beweisschema wird anhand der eingesreuten Annotationen in P deutlich:

```
1    i, s  var Nat;

2    { true }
3    i, s := 1, b[0];
4    { I }
5    do (i < 11) then
6        { (i < 11) ∧ I }
7        i, s := i + 1, s + b[i];
8        { I }
9    od
10   { (i ≥ 11) ∧ I }
11   { R :  s = Σ_{k: 0≤k<11} b[k] }
```

Prinzipiell muss gezeigt werden:

($*$) Das Prädikat I gilt zu Beginn der do-Schleife (Zeile 4).

($**$) Ein einzelner Schleifendurchlauf verändert das Prädiakt I nicht. I gilt also direkt vor und direkt nach jeder Iteration der Schleife (Zeile 6 auf 8).

($***$) Somit gilt I auch nach der Schleife und es bleibt noch zu zeigen, dass aus I und dem Nichtzutreffen aller Wächter der Schleife die gewünschte Nachbedingung R folgt (Zeile 10 auf 11).

Insgesamt gilt dann: Sofern P terminiert, gilt die gewünschte Nachbedingung R. Technisch erfolgen die einzelnen Beweise durch Berechnung der jeweiligen Schwächsten Vorbedingung wp(...).

(b) Das Prädikat I gilt sowohl vor Beginn der Schleife als auch nach jeder Schleifeniteration:

$$I \quad \equiv \quad \Big((1 \le i \le 11) \ \wedge \ (s = \sum\nolimits_{k:\ 0 \le k < i} b[k])\Big)$$

(c) Zu zeigen $(*)$: I gilt zu Beginn der Schleife:

$$\begin{aligned}\mathsf{wp}(i, s \;:=\; 1, b[0],\;\; I) &= (1 \leq 1 \leq 11) \;\wedge\; (b[0] = \textstyle\sum_{k:\ 0 \leq k < 1} b[k]) \\ &= \mathit{true} \;\wedge\; (b[0] = b[0]) \\ &= \mathit{true}\end{aligned}$$

Somit gilt I zu Beginn der Schleife für jede mögliche Belegung der Variablen (true).

Zu zeigen $(**)$: I bleibt während eines Schleifendurchlaufs gültig:

$$\begin{aligned}\mathsf{wp}(i, s \;:=\; i+1, s+b[i],\;\; I) &= (1 \leq i+1 \leq 11) \;\wedge\; (s + b[i] = \textstyle\sum_{k:\ 0 \leq k < i+1} b[k]) \\ &= (0 \leq i < 11) \;\wedge\; (s + b[i] = \textstyle\sum_{k:\ 0 \leq k < i} b[k] + b[i]) \\ &= (0 \leq i < 11) \;\wedge\; (s = \textstyle\sum_{k:\ 0 \leq k < i} b[k])\end{aligned}$$

Obige Schwächste Vorbedingung wird durch das Prädikat $\big(I \;\wedge\; (i < 11)\big)$ impliziert.

Zu zeigen $(***)$: $\big(I \wedge (i \geq 11)\big) \;\Rightarrow\; R$
Gilt offensichtlich.

Aufgabe 2 Prädikative Spezifikation [LÖSUNG]

Für das Programm P:

```
x, y, z := a, b, 1;
do y > 0 then
    if odd(y)    then y := y − 1; z := z ∗ x
     [] even(y) then x := x ∗ x; y := y ÷ 2
    fi
od
```

ergibt sich folgenden Prädikative Spezifikation:

$Pre =_{def}$ **true**

$Post =_{def} z = a^b$

Schrittweise Berechnung von $PS[P]$ durch einfaches Anwenden der Regeln für PS (Es wird davon ausgegangen, dass alle Auswertungen in P terminieren):

$$\begin{aligned}PS[P] &= PS[(x, y, z) := (a, b, 1);\; DO] \\ &= PS[(x, y, z) := (a, b, 1)][x/\acute{x},\; y/\acute{y},\; z/\acute{z}] \;\wedge\; PS[DO][x/\grave{x},\; y/\grave{y},\; z/\grave{z}] \\ &= PS[(x, y, z) := (a, b, 1)] \;\wedge\; PS[DO][x/\grave{x},\; y/\grave{y},\; z/\grave{z}] \\ &= (\acute{x} = a \;\wedge\; \acute{y} = b \;\wedge\; \acute{z} = 1) \;\wedge\; PS[DO][x/\grave{x},\; y/\grave{y},\; z/\grave{z}]\end{aligned}$$

$$
\begin{aligned}
PS[DO] &= PS[\textbf{if } y > 0 \textbf{ then } IF;\ DO \\
&\qquad [] \ y \leq 0 \textbf{ then nop fi}] \\
&= \{\text{Definition von } PS[\textbf{if}...\textbf{fi}]\} \\
&\quad \big(\neg(\grave{y} > 0) \ \wedge \ \neg(\grave{y} \leq 0)\big) \ \vee \\
&\quad \big((\grave{y} > 0) \ \wedge \ PS[IF][x/\acute{x},\ y/\acute{y},\ z/\acute{z}] \ \wedge \ PS[DO][x/\grave{x},\ y/\grave{y},\ z/\grave{z}]\big) \ \vee \\
&\quad \big((\grave{y} \leq 0) \ \wedge \ (\acute{x} = \grave{x}) \ \wedge \ (\acute{y} = \grave{y}) \ \wedge \ (\acute{z} = \grave{z})\big) \\
&= \mathit{false} \ \vee \\
&\quad \big((\grave{y} > 0) \ \wedge \ PS[IF] \ \wedge \ PS[DO][x/\grave{x},\ y/\grave{y},\ z/\grave{z}]\big) \ \vee \\
&\quad \big((\grave{y} \leq 0) \ \wedge \ (\acute{x} = \grave{x}) \ \wedge \ (\acute{y} = \grave{y}) \ \wedge \ (\acute{z} = \grave{z})\big)
\end{aligned}
$$

$$
\begin{aligned}
PS[IF] &= PS[\textbf{if } odd(y) \textbf{ then } y := y - 1; z := z * x \\
&\qquad [] \ even(y) \textbf{ then } x := x * x; y := y \div 2 \textbf{ fi}] \\
&= \{\text{Definition von } PS[\textbf{if}...\textbf{fi}]\} \\
&\quad \big(\neg(odd(\grave{y}) \ \wedge \ \neg(even(\grave{y})\big) \ \vee \\
&\quad \big(odd(\grave{y}) \ \wedge \ PS[y := y - 1;\ z := z * x]\big) \ \vee \\
&\quad (even(\grave{y}) \ \wedge \ PS[x := x * x;\ y := y \div 2]\big) \ \vee \\
&= \mathit{false} \ \vee \\
&\quad \big(odd(\grave{y}) \ \wedge \ (\acute{x} = \grave{x}) \ \wedge \ (\acute{y} = \grave{y} - 1) \ \wedge \ (\acute{z} = \grave{z} * \grave{x})\big) \ \vee \\
&\quad \big(even(\grave{y}) \ \wedge \ (\acute{x} = \grave{x} * \grave{x}) \ \wedge \ (\acute{y} = \grave{y} \div 2) \ \wedge \ (\acute{z} = \grave{z})\big)
\end{aligned}
$$

Aufgabe 3 Anweisungsorientierte Programme und Zusicherungen [LÖSUNG]

Annotiertes Anweisungsorientiertes Programm für die Berechnung: $v = \frac{(x+y)*z}{x+y+z}$

1 $PRE = \{x = X \wedge y = Y \wedge z = Z \wedge X + Y + Z \neq 0\}$

2

3 $var\ a, b, c : Int$

4

5 $a := x + y;\ \{x = X \wedge y = Y \wedge z = Z \wedge X + Y + Z \neq 0 \wedge a = X + Y\}$

6 $b := a * z;\ \{x = X \wedge y = Y \wedge z = Z \wedge X + Y + Z \neq 0 \wedge a = X + Y$
$\wedge\ b = (X + Y) * Z\}$

7 $c := a + z;\ \{x = X \wedge y = Y \wedge z = Z \wedge X + Y + Z \neq 0 \wedge a = X + Y$
$\wedge\ b = (X + Y) * Z \wedge c = X + Y + Z\}$

8 $v := b \div c;\ \{x = X \wedge y = Y \wedge z = Z \wedge X + Y + Z \neq 0 \wedge a = X + Y$
$\wedge\ b = (X + Y) * Z \wedge c = X + Y + Z \wedge v = \frac{(X+Y)*Z}{X+Y+Z}\}$

9

10 $POST = \{x = X \wedge y = Y \wedge z = Z \wedge X + Y + Z \neq 0 \wedge v = \frac{(X+Y)*Z}{X+Y+Z}\}$

Aufgabe 4 Zusicherungsbeweis [LÖSUNG]

Zusicherungsbeweis für das Programm aus Aufgabe 3.
Die Prädikate Rxy charakterisieren dabei die Zustände zwischen den Zeilen x und y.

$\Rightarrow$ Kompositions-Regel:

$\{R87\}\ \ v := b \div c\ \ \{POST\}$

$\{R76\}\ \ c := a + z\ \ \{R87\}$

$\{R65\}\ \ b := a * z\ \ \{R76\}$

$\{PRE\}\ \ a := x + y\ \ \{R65\}$

$\Rightarrow$ Zuweisungs-Regel:

$R78 = POST[b \div c/v]$

$R67 = R78[(a + z)/c]$

$R56 = R67[(a * z)/b]$

$PRE = R56[(x + y)/a]$

Aufgabe 5 Vollständige Vor- bzw. Nachbedingungen [LÖSUNG]

Betrachte folgendes Programm *Prog* zur Berechnung der Fakultät:

$y := 1;$
do $(x \neq 0)$ **then** $y := y * x;$
$x := x - 1;$
od

Das Hoare-Triple $\{x \geq 0\}\ Prog\ \{y = x!\}$ ist zur Beschreibung des Programms *Prog* nicht geeignet, da die Variable x am Ende von *Prog* den Wert 0 hat (*Prog* "konsumiert" x während der Ausführung). Vielmehr meint man mit x in obiger Nachbedingung den intialen Wert der Variable x. Ein geignetes Hoare-Tripel wäre also $\{(x = x_0) \wedge (x \geq 0)\}\ Prog\ \{y = x_0!\}$, wobei x_0 den initialen Wert der Variable x enthält (x_0 nennt man dabei eine *Logische Variable*.).

Für die Fragestellung bedeutet das also, dass eine Anweisung S unter der Annahme $\{x = y\}\ S\ \{x = y!\}$ die Fakultätsfunktion nur dann berechnet, falls der Wert von y vor und nach Ausführung von S identisch ist (y entspricht dabei einer logischen Variable; vgl. die Variable x_0 aus *Prog*). Für $y \leq 0$ kann davon ausgegangen werden, dass das Programm nicht konvergiert. Versteht man unter der Fragestellung die totale Korrektheit, so ist zusätzlich die Terminierung von S zu fordern.

Aufgabe 6 *wp*-**Kalkül [LÖSUNG]**

Es ergeben sich folgende Vorbedingungen:

- $\mathsf{wlp}(S, x = y) \;=\; true$
  Falls die **do**-Schleife terminiert gilt auf jeden Fall die Negation des Wächters, also hier $\neg(x \neq y) \equiv (x = y) \equiv Q$, d.h. falls S überhaupt terminiert ist die Nachbedingung Q sowieso immer erfüllt, insbesondere für jeden Startzustand (ausgedrückt durch das Prädikat $true$).
- $\mathsf{wp}(S, x \neq y) \;=\; false$
  Die Schleife (und damit S) terminiert nur, falls der Wächter nicht mehr gilt. Insbesondere gilt damit bei Terminierung von S das Prädikat $(x = y)$ (Negation des einzigen Wächters der **do**-Schleife). Somit gibt es keinen Zustand, so dass die geforderte Nachbedingung $(x \neq y)$ erfüllt werden kann (ausgedrückt durch das Prädikat $false$).
- $\mathsf{wlp}(S, x \neq y) \;=\; x > y$
- $\mathsf{wp}(S, x = y) \;=\; x \leq y$