3. Globalübung Programmierung (13.11.07)

Aufgabe 1

a) Verifizieren Sie die partielle Korrektheit des folgenden Algorithmus P mit Hilfe des Hoare-Kalküls. Hierbei dürfen zwei Zusicherungen nur dann direkt untereinander stehen, wenn die untere aus der oberen folgt. Hinter einer Programmanweisung darf nur eine Zusicherung stehen, wenn dies aus einer Regel des Hoare-Kalküls folgt. Kennzeichnen Sie Ihre benutzte Invariante deutlich.

Eingabe:	n
Ausgabe:	res
Vorbedingung:	$n \geq 0$
Nachbedingung:	$res = 2 * n$

$i = 0;$
$res = 0;$
`while` $(i < n)$ $\{$
$\quad res = res + 2;$
$\quad i = i + 1;$
$\}$

b) Beweisen Sie die Terminierung des Algorithmus P. Geben Sie hierzu eine Variante für die `while`-Schleife an. Zeigen Sie, dass es sich tatsächlich um eine Variante handelt und beweisen Sie damit unter Verwendung des Hoare-Kalküls mit der Voraussetzung $n \geq 0$ die Terminierung.

Lösungsvorschlag:

a) $\langle n \geq 0 \rangle$

$\langle n \geq 0 \wedge 0 = 0 \rangle$

$i = 0;$

$\langle n \geq 0 \wedge i = 0 \rangle$

$\langle n \geq 0 \wedge i = 0 \wedge 0 = 0 \rangle$

$res = 0;$

$$\langle n \geq 0 \wedge i = 0 \wedge res = 0 \rangle$$

$$\langle i \leq n \wedge res = 2 * i \rangle$$

`while` $(i < n)$ $\{$

$$\langle i \leq n \wedge res = 2 * i \wedge i < n \rangle$$

$$\langle i < n \wedge res + 2 = 2 * i + 2 \rangle$$

$res = res + 2;$

$$\langle i < n \wedge res = 2 * i + 2 \rangle$$

$$\langle i + 1 \leq n \wedge res = 2 * (i + 1) \rangle$$

$i = i + 1;$

$$\langle i \leq n \wedge res = 2 * i \rangle$$

$\}$

$$\langle i \leq n \wedge res = 2 * i \wedge i \not< n \rangle$$

$$\langle res = 2 * n \rangle$$

Die hier verwendete Schleifeninvariante ist:

$$\langle i \leq n \wedge res = 2 * i \rangle$$

Man kann die Invariante z.B. dadurch finden, dass man die jeweiligen Werte der vorkommenden Variablen bei der Überprüfung der Schleifenbedingung für eine sinnvoll gewählte Eingabe in eine Tabelle einträgt und daraus einen Zusammenhang zwischen den Variablen ersieht. In diesem Beispiel gilt für $n = 4$:

n	i	res
4	0	0
4	1	2
4	2	4
4	3	6
4	4	8

Zum einen sieht man, dass für die gewählte Eingabe res immer den doppelten Wert von i hat, also $res = 2 * i$. Zum anderen gilt hier immer $i \leq n$. Dies ist wichtig dafür, eine Aussage für den Wert von i bei Verlassen der Schleife machen zu können (siehe Schritt von der vorletzten zur letzten Zusicherung).

Alternativ kann man die Schleifeninvariante natürlich auch durch „scharfes Hinsehen“ finden und dadurch, dass man sich überlegt, wie der in der Nachbedingung geforderte Zusammenhang in den Schleifendurchläufen mit den verwendeten Variablen hergestellt wird.

b) Eine Variante ist $V = n - i$, denn aus der Schleifenbedingung $i < n$ folgt $n - i \geq 0$. Somit:

$$\langle n - i = m \wedge i < n \rangle$$

$$\langle n - (i + 1) < m \rangle$$

$res = res + 2;$

$$\langle n - (i + 1) < m \rangle$$

$i = i + 1;$

$$\langle n - i < m \rangle$$

Damit ist die Terminierung der einzigen Schleife in P gezeigt.

Aufgabe 2

a) Verifizieren Sie die partielle Korrektheit des folgenden Algorithmus P mit Hilfe des Hoare-Kalküls. Hierbei dürfen zwei Zusicherungen nur dann direkt untereinander stehen, wenn die untere aus der oberen folgt. Hinter einer Programmanweisung darf nur eine Zusicherung stehen, wenn dies aus einer Regel des Hoare-Kalküls folgt. Kennzeichnen Sie Ihre benutzte Invariante deutlich.

Eingabe:	Array a der Länge n von ganzen Zahlen
Ausgabe:	res
Vorbedingung:	$n \geq 0$
Nachbedingung:	$res = \sum_{j=0}^{n-1} a[j]$

$res = 0;$
$i = n;$
`while` $(i > 0)$ $\{$
  $i = i - 1;$

$res = res + a[i];$
}

b) Beweisen Sie die Terminierung des Algorithmus P. Geben Sie hierzu eine Variante für die `while`-Schleife an. Zeigen Sie, dass es sich tatsächlich um eine Variante handelt und beweisen Sie damit unter Verwendung des Hoare-Kalküls mit der Voraussetzung $n \geq 0$ die Terminierung.

Lösungsvorschlag:

a)

$$\langle n \geq 0 \rangle$$

$$\langle n \geq 0 \land 0 = 0 \rangle$$

$res = 0;$

$$\langle n \geq 0 \land res = 0 \rangle$$

$$\langle n \geq 0 \land res = 0 \land n = n \rangle$$

$i = n;$

$$\langle i \geq 0 \land res = 0 \land i = n \rangle$$

$$\langle i \geq 0 \land res = \textstyle\sum_{j=i}^{n-1} a[j] \land i \leq n \rangle$$

`while` $(i > 0)$ {

$$\langle i \geq 0 \land res = \textstyle\sum_{j=i}^{n-1} a[j] \land i \leq n \land i > 0 \rangle$$

$$\langle i - 1 \geq 0 \land res = \textstyle\sum_{j=(i-1)+1}^{n-1} a[j] \land i - 1 < n \rangle$$

$i = i - 1;$

$$\langle i \geq 0 \land res = \textstyle\sum_{j=i+1}^{n-1} a[j] \land i < n \rangle$$

$$\langle i \geq 0 \land res + a[i] = \textstyle\sum_{j=i}^{n-1} a[j] \land i \leq n \rangle$$

$res = res + a[i];$

$$\langle i \geq 0 \land res = \textstyle\sum_{j=i}^{n-1} a[j] \land i \leq n \rangle$$

}

$$\langle i \geq 0 \wedge res = \textstyle\sum_{j=i}^{n-1} a[j] \wedge i \leq n \wedge i \not> 0 \rangle$$

$$\langle res = \textstyle\sum_{j=0}^{n-1} a[j] \rangle$$

Die hier verwendete Schleifeninvariante ist:

$$\langle i \geq 0 \wedge res = \sum_{j=0}^{n-1} a[j] \wedge i \leq n \rangle$$

In diesem Beispiel gilt für die jeweiligen Werte der Variablen bei der Überprüfung der Schleifenbedingung für $n = 4$:

n	i	res
4	4	0
4	3	$a[3]$
4	2	$a[3] + a[2]$
4	1	$a[3] + a[2] + a[1]$
4	0	$a[3] + a[2] + a[1] + a[0]$

b) Eine Variante ist $V = i$, denn aus der Schleifenbedingung $i > 0$ folgt $i \geq 0$. Somit:

$$\langle i = m \wedge i > 0 \rangle$$

$$\langle i - 1 < m \rangle$$

$i = i - 1;$

$$\langle i < m \rangle$$

$res = res + a[i];$

$$\langle i < m \rangle$$

Damit ist die Terminierung der einzigen Schleife in P gezeigt.

Aufgabe 3

a) Verifizieren Sie die partielle Korrektheit des folgenden Algorithmus P mit Hilfe des Hoare-Kalküls. Hierbei dürfen zwei Zusicherungen nur dann direkt untereinander stehen, wenn die untere aus der oberen folgt. Hinter einer Programmanweisung darf nur eine Zusicherung stehen, wenn dies aus einer Regel des Hoare-Kalküls folgt. Kennzeichnen Sie Ihre benutzte Invariante deutlich.

Eingabe: x, y
Ausgabe: res
Vorbedingung: $x > 0 \wedge y \geq 0$
Nachbedingung: $res = x^y$

$z = y;$
$res = 1;$
`while` $(z > 0)$ {
  $res = res * x;$
  $z = z - 1;$
}

b) Beweisen Sie die Terminierung des Algorithmus P. Geben Sie hierzu eine Variante für die `while`-Schleife an. Zeigen Sie, dass es sich tatsächlich um eine Variante handelt und beweisen Sie damit unter Verwendung des Hoare-Kalküls mit der Voraussetzung $x > 0 \wedge y \geq 0$ die Terminierung.

Lösungsvorschlag:

a)

$$\langle x > 0 \wedge y \geq 0 \rangle$$

$$\langle y \geq 0 \wedge y = y \rangle$$

$z = y;$

$$\langle y \geq 0 \wedge z = y \rangle$$

$$\langle z \geq 0 \wedge z = y \wedge 1 = 1 \rangle$$

$res = 1;$

$$\langle z \geq 0 \wedge z = y \wedge res = 1 \rangle$$

$$\langle z \geq 0 \wedge res = x^{y-z} \rangle$$

`while` $(z > 0)$ {

$$\langle z \geq 0 \wedge res = x^{y-z} \wedge z > 0 \rangle$$

$$\langle z > 0 \wedge res * x = x^{y-(z-1)} \rangle$$

  $res = res * x;$

$$\langle z > 0 \wedge res = x^{y-(z-1)} \rangle$$

$$\langle z - 1 \geq 0 \wedge res = x^{y-(z-1)} \rangle$$

$z = z - 1;$

$$\langle z \geq 0 \wedge res = x^{y-z} \rangle$$

}

$$\langle z \geq 0 \wedge res = x^{y-z} \wedge z \not> 0 \rangle$$

$$\langle res = x^y \rangle$$

Die hier verwendete Schleifeninvariante ist:

$$\langle z \geq 0 \wedge res = x^{y-z} \rangle$$

In diesem Beispiel gilt für die jeweiligen Werte der vorkommenden Variablen bei der Überprüfung der Schleifenbedingung für $x = 2 \wedge y = 4$:

x	y	z	res
2	4	4	1
2	4	3	2
2	4	2	4
2	4	1	8
2	4	0	16

b) Eine Variante ist $V = z$, denn aus der Schleifenbedingung $z > 0$ folgt $z \geq 0$. Somit:

$$\langle z = m \wedge z > 0 \rangle$$

$$\langle z - 1 < m \rangle$$

$res = res * x;$

$$\langle z - 1 < m \rangle$$

$z = z - 1;$

$$\langle z < m \rangle$$

Damit ist die Terminierung der einzigen Schleife in P gezeigt.