Dr. Theo Lettmann

Paderborn, den 6. Dezember
Abgabe 15. Dezember 2002

Übungen zur Vorlesung

Modellierung

WS 2003/2004

Blatt 8

Vorbemerkung:
Gehen Sie in den nachfolgenden Aufgaben davon aus, dass sämtliche Variablen ganzzahlig sind.

AUFGABE 49 :
Überprüfen Sie die folgenden Verifikationsschritte auf ihre Korrektheit, indem Sie Zwischenschritte einfügen.

a)
$\{a = b^3\}$
$a := a - 3 * b * b + 3 * b - 1;$
$b := b - 1;$
$\{a = b^3\}$

b)
$\{a \cdot b \geq 0\}$
if $(a > b)$ *then begin*
$x := a - b;$
$b := b - a;$
$a := x;$
end
else
$a := b;$
$\{a \cdot b < 0\}$

Lösung:

a) Korrekt, denn es gilt:

$\{a = b^3\}$
$\rightarrow \{a - 3b^2 + 3b - 1 = b^3 - 3b^2 + 3b - 1\}$
$a := a - 3 * b * b + 3 * b - 1;$
$\{a = b^3 - 3b^2 + 3b - 1\}$
$\rightarrow \{a = (b-1)^3\}$
$b := b - 1;$
$\{a = b^3\}$

b) Falsch, denn es gilt:

$\{a \cdot b \geq 0\}$
if $(a > b)$ *then begin*
$\{a \cdot b \geq 0$ und $a > b\}$
$\rightarrow \{a \cdot b \geq 0$ und $a > b$ und $a - b = a - b\}$
$x := a - b;$

$\{a \cdot b \geq 0$ und $a > b$ und $x = a - b\}$
$\rightarrow \{a \cdot (b - a + a) \geq 0$ und $0 > b - a$ und $-x = b - a\}$
$b := b - a;$
$\{a \cdot (b + a) \geq 0$ und $0 > b$ und $-x = b\}$
$\rightarrow \{0 > b$ und $-x = b$ und $x = x\}$
$a := x;$
$\{0 > b$ und $-x = b$ und $a = x\}$
$\rightarrow \{a \cdot b \geq 0\}$
end
else
$\{a \cdot b \geq 0$ und $a \leq b\}$
$\rightarrow \{b = b\}$
$a := b;$
$\{a = b\}$
$\rightarrow \{a \cdot b \geq 0\}$
$\{a \cdot b \geq 0\}$
$\not\rightarrow \{a \cdot b < 0\}$

AUFGABE 50 :
Ermitteln Sie zu den folgenden Programmausschnitten jeweils die fehlende Vor- bzw. Nachbedingung. Geben Sie in ihrer Lösung alle Zwischenschritte an und notieren Sie die benutzte Regel.

a)
$\{\ldots\}$
$x := x + 3;$
$y := x - 2 * z;$
$\{x > 3$ und $3y > x\}$

b)
$\{a + 1 > 0\}$
if $(b < 0)$ *then*
$a := -4 * b;$
else
$a := 4 * b;$
$\{\ldots\}$

Lösung:

a)
$\{x > 0$ und $x + 3 > 3 * z\}$
$\rightarrow \{x > 0$ und $2(x + 3) - 6 * z) > 0\}$ Abschwächungsregel
$\rightarrow \{x + 3 > 3$ und $3(x + 3 - 2 * z) > x + 3\}$ Abschwächungsregel
$x := x + 3;$
$\{x > 3$ und $3(x - 2 * z) > x\}$ Zuweisungsregel
$y := x - 2 * z;$
$\{x > 3$ und $3y > x\}$ Zuweisungsregel

Als Vorbedingung wäre auch schon die Zusicherung von Zeile 3 korrekt gewesen.

b)

$\{a+1>0\}$
if $(b<0)$ *then*
$\{a+1>0$ und $b<0\}$ Regel für 2-seitige bedingte Anweisung
$\rightarrow \{b<0$ und $-4*b=-4*b\}$ Abschwächungsregel
$a := -4*b;$
$\{b<0$ und $a=-4*b\}$ Zuweisungsregel
$\rightarrow \{a=4*|b|\}$ Abschwächungsregel
else
$\{a+1>0$ und $b\geq 0\}$ Regel für 2-seitige bedingte Anweisung
$\rightarrow \{b\geq 0$ und $4*b=4*b\}$ Abschwächungsregel
$a := 4*b;$
$\{b\geq 0$ und $a=4*b\}$ Zuweisungsregel
$\rightarrow \{a=4*|b|\}$ Abschwächungsregel
$\{a=4*|b|\}$ Regel für 2-seitige bedingte Anweisung
$\rightarrow \{a\geq 0\}$ Abschwächungsregel

KORREKTURAUFGABE 51 (3 Punkte) **:**
Verifizieren Sie den nachfolgenden Programmausschnitt. Was tut das Programm?

$\{a=x$ und $b=y\}$
begin
$a := a-b;$
$b := a+b;$
$a := b-a;$
end
$\{\ldots\}$

Lösung:

$\{a=x$ und $b=y\}$
begin
$\{a=x$ und $b=y\}$
$\rightarrow \{a-b+b=x$ und $b=y\}$
$a := a-b;$
$\{a+b=x$ und $b=y\}$
$\rightarrow \{a+b=x$ und $a+b=a+y\}$
$b := a+b;$
$\{b=x$ und $b=a+y\}$
$\rightarrow \{b=x$ und $b-a=y\}$
$a := b-a;$
$\{b=x$ und $a=y\}$
end
$\{b=x$ und $a=y\}$

Das Programm vertauscht die Werte der zwei Variablen a und b.

AUFGABE 52 :
Gegeben sei das folgende Programm:

$\{n \geq 0\}$
begin
  $i := 0;$
  $x := 1;$
  while $(i < n)$ *do begin*
    $i := i + 1;$
    $x := i * x;$
  end
end
$\{\ldots\}$

a) Bestimmen Sie, was das Programm berechnet und leiten Sie daraus eine Spezifikation ab.

b) Vervollständigen Sie das Programm um die einzelnen Verifikationsschritte. Dabei sei eine mögliche Invariante durch $\{x = i!$ und $i \leq n\}$ gegeben.

c) Zeigen Sie, dass es sich bei $\{x = n!\}$ um keine Invariante für die Schleife handelt.

d) Begründen Sie, warum $\{true\}$ und $\{false\}$ keine guten Invarianten für die Schleife darstellen.

Lösung:

a) Das Programm berechnet die Fakultät von n, so dass eine Spezifikation gegeben ist durch

Vorbedingung: $\{n \geq 0\}$
Nachbedingung: $\{x = n!\}$

b)
$\{n \geq 0\}$
begin
  $\{n \geq 0\} \rightarrow \{n \geq 0$ und $0 = 0\}$
  $i := 0;$
  $\{n \geq 0$ und $i = 0\} \rightarrow \{n \geq 0$ und $i = 0$ und $1 = 1\}$
  $x := 1;$
  $\{n \geq 0$ und $i = 0$ und $x = 1\}$
  $\rightarrow \{x = i!$ und $i \leq n\}$
  while $(i < n)$ *do begin*
    $\{x = i!$ und $i \leq n$ und $i < n\}$
    $\rightarrow \{x = (i + 1 - 1)!$ und $i + 1 - 1 < n\}$
    $i := i + 1;$
    $\{x = (i - 1)!$ und $i - 1 < n\}$
    $\rightarrow \{i * x = (i - 1)! * i$ und $i - 1 < n\}$
    $x := i * x;$
    $\{x = (i - 1)! * i$ und $i - 1 < n\}$
    $\rightarrow \{x = i!$ und $i \leq n\}$
  end

$\{x = i! \text{ und } i \leq n \text{ und } i \geq n\}$
$\rightarrow \{x = i! \text{ und } i = n\}$
$\rightarrow \{x = n!\}$

end
$\{x = n!\}$

c)
$\{x = n!...\}$
while $(i < n)$ *do begin*
$\{x = n!... \text{ und } i < n\}$
$\rightarrow \{x = n!... \text{ und } i + 1 - 1 < n\}$
$i := i + 1;$
$\{x = n!... \text{ und } i - 1 < n\}$
$\rightarrow \{i * x = n! * i... \text{ und } i - 1 < n\}$
$x := i * x;$
$\{x = n! * i... \text{ und } i - 1 < n\}$
$\rightarrow \{x = n! * i... \text{ und } i \leq n\}$
end

Gilt $\{x = n!\}$ wie eine Invariante vor der Schleife, so muss sich die nach dem Durchlauf durch den Schleifenrumpf ergebende Zusicherung $\{x = n! * i \text{ und } i \leq n\}$ abschwächen lassen zu $\{x = n!\}$. Dies ist aber nur möglich, wenn $i = 1$ am Ende des Schleifenrumpfes gilt, also nur für $n = 1$. In allen anderen Fällen ändert sich x beim Durchlaufen des Schleifenrumpfes, n jedoch nicht, so dass die Zusicherung $\{x = n!\}$ am Ende des zweiten Durchlaufs durch den Schleifenrumpf nicht mehr gültig ist. $\{x = n!\}$ ist daher keine Invariante.

d) $\{true\}$ ist eine zu schwache Invariante. $\{true\}$ gilt zwar an allen nötigen Stellen der Schleife, dies gilt jedoch für jede beliebige Schleife, so dass sie zur formalen Verifikation ziemlich ungeignet ist. Das Gegenteil gilt für $\{false\}$, denn diese Invariante ist so stark, dass alle Zusicherungen daraus gefolgert werden können.

Die Zusicherung $\{false\}$ kann nur auftreten, wenn ein Programmzweig niemals durchlaufen wird, z.B. der *else*-Zweig einer bedingten Anweisung.

e) Die Schleife terminiert, da sich der Wert der ganzzahligen Variablen i in jedem Durchlauf um 1 erhöht wird und i zudem nach oben durch n beschränkt ist ($i \leq n$ ist Teil der Invarianten).

AUFGABE 53 :
Gegeben sei das folgende Programm:

$\{n \geq 0\}$
begin
$i := 0;$
$x := 0;$
while $(i < n)$ *do begin*
$i := i + 1;$
$x := x + i;$
end
end
$\{x = \sum_{i=0}^{n} i\}$

a) Vervollständigen Sie das Programm um die einzelnen Verifikationsschritte. Wählen Sie dazu eine Kombination aus zwei der folgenden Zusicherungen:
$I_1 = \{n \geq i\}$, $I_2 = \{2x = i^2\}$, $I_3 = \{x = \frac{i(i+1)}{2}\}$, $I_4 = \{i \geq n\}$.

b) Zeigen Sie, dass das Programm terminiert.

Lösung:

(a) Verifikation des Anfangsstückes

$\{n \geq 0\}$
begin
  $\{n \geq 0\}$
  $\rightarrow \{n \geq 0, 0 = 0\}$
  $i := 0;$
  $\{n \geq 0, i = 0\}$
  $\rightarrow \{n \geq 0, i = 0, 0 = 0\}$
  $x := 0;$
  $\{n \geq 0, i = 0, x = 0\}$
  while $(i < n)$ *do begin*
    $i := i + 1;$
    $x := x + i;$
  end
end

Test der Zusicherung $2x = i^2$

$\{n \geq 0\}$
begin
  $i := 0;$
  $x := 0;$
  $\{n \geq 0, i = 0, x = 0\}$
  $\rightarrow \{2x = i^2, ...\}$
  while $(i < n)$ *do begin*
    $\{2x = i^2, ..., n > i\}$
    $\rightarrow \{2x = (i + 1 - 1)^2, n \geq i + 1, ...\}$
    $i := i + 1;$
    $\{2x = (i - 1)^2, n \geq i, ...\}$
    $\rightarrow \{2(x + i - i) = (i - 1)^2, n \geq i, ...\}$
    $x := x + i;$
    $\{2(x - i) = (i - 1)^2, n \geq i, ...\}$
    $\rightarrow \{2x - 2i = i^2 - 2i + 1, n \geq i, ...\}$
    $\not\rightarrow \{2x = i^2, ...\}$
  end
end

Also ist Zusicherung $2x = i^2$ als Invariante NICHT geeignet.

Test der Zusicherung $i \geq n$

$\{n \geq 0\}$
begin
$i := 0;$
$x := 0;$
$\{n \geq 0, i = 0, x = 0\}$
$\not\rightarrow \{i \geq n, ...\}$
while $(i < n)$ *do begin*
$i := i + 1;$
$x := x + i;$
end
end

Also ist Zusicherung $i \geq n$ als Invariante NICHT geeignet.

Die anderen beiden Zusicherungen sind als Invariante geeignet:

Fortsetzung der Verifikation mit Invarianter $n \geq i, x = \frac{1}{2}i(i+1)$

$\{n \geq 0\}$
begin
$\{n \geq 0\}$
$\rightarrow \{n \geq 0, 0 = 0\}$
$i := 0;$
$\{n \geq 0, i = 0\}$
$\rightarrow \{n \geq 0, i = 0, 0 = 0\}$
$x := 0;$
$\{n \geq 0, i = 0, x = 0\}$
$\rightarrow \{n \geq i, x = \frac{1}{2}i(i+1)\}$
while $(i < n)$ *do begin*
$\{n \geq i, x = \frac{1}{2}i(i+1), n > i\}$
$\rightarrow \{x = \frac{1}{2}(i+1-1)(i+1), n \geq i+1\}$
$i := i + 1;$
$\{x = \frac{1}{2}(i-1)i, n \geq i\}$
$\rightarrow \{x + i - i = \frac{1}{2}(i-1)i, n \geq i\}$
$x := x + i;$
$\{x - i = \frac{1}{2}(i-1)i, n \geq i\}$
$\rightarrow \{x = \frac{1}{2}((i-1)i + 2i), n \geq i\}$
$\rightarrow \{x = \frac{1}{2}i(i+1), n \geq i\}$
end
$\{x = \frac{1}{2}i(i+1), n \geq i, i \geq n\}$
$\rightarrow \{x = \frac{1}{2}i(i+1), n = i\} \rightarrow \{x = \frac{1}{2}n(n+1)\}$
end
$\{x = \frac{1}{2}n(n+1)\}$
$\rightarrow \{x = \sum_{i=0}^{n} i\}$

Damit ist die partielle Korrektheit gezeigt.

b) Terminierung

$\{n \geq 0\}$
begin
$\{n \geq 0\}$
$\rightarrow \{n \geq 0, 0 = 0\}$
$i := 0;$
$\{n \geq 0, i = 0\}$
$\rightarrow \{n \geq 0, i = 0, 0 = 0\}$
$x := 0;$
$\{n \geq 0, i = 0, x = 0\}$

while $(i < n)$ *do begin*
$i := i + 1;$
$x := x + i;$
end
end
$\{x = \sum_{i=0}^{n} i\}$

Kandidaten für Variante T:
i, x oder arithmetische Ausdrücke hiermit, da nur sie in der Schleife verändert werden.

Kandidat $T = i$:

- Nachweis der Ganzzahligkeit: Anfangswerte und Veränderungen
  T ganzzahlig, da mit 0 initialisiert in $i := 0;$ und nur durch $i := i + 1;$ in der Schleife verändert.
- Nachweis der Inkrementierung von T und Beschränktheit von T für jeden Schleifendurchlauf
  Verifikation von S mit Vorbedingung $\{T = k, T \leq ub, I, B\}$ zur Nachbedingung $\{T = k + 1, T \leq ub, I\}$
  (k bezeichnet eine Variable, die nicht im Programm vorkommt)

  while $(i < n)$ *do begin*
  $\{i = k, n \geq i, n > i\}$
  $\rightarrow \{i + 1 = k + 1, n \geq i + 1\}$
  $i := i + 1;$
  $\{i = k + 1, n \geq i\}$
  $x := x + i;$
  $\{i = k + 1, n \geq i\}$
  end

- Aus Zusicherung $\{T \geq ub, I\}$ durch Abschwächung die Zusicherung $\{\text{nicht } B\}$ erzeugen.
  $\{i \geq n\} \rightarrow \{\text{nicht } i < n\}$

Damit folgt insgesamt, dass die Schleife terminiert.

Alternative Begründung:
Diese Begründung ist im Teil der Inkrementierung von T weniger formal und auch im Bereich der Folgerbarkeit der Ungültigkeit der Schleifenbedingung bei Überschreiten der Schranke.

Kandidat $T = i$:

- T ganzzahlig, da mit 0 initialisiert in $i := 0;$ und nur durch $i := i + 1;$ in der Schleife verändert.
- T ist wächst mit jedem Durchlauf durch den Schleifenrumpf um den Wert 1, da i im Schleifenrumpf nur durch $i := i + 1;$ verändert wird.
- T ist nach oben beschränkt, da $n \geq i$ Teil der Invariante aus dem Nachweis der partiellen Korrektheit ist, also aus $n \geq i$ und $n > i$ (Schleifenbedingung) als Vorbedingung im Schleifenrumpf wieder $n \geq i$ als Nachbedingung des Schleifenrumpfes folgt.

  Da die Zusicherung $n \geq i$ als Teil der Invarianten auch vor der Schleife richtig ist, liegt T auch vor der Schleife unter der Schranke.

Damit folgt insgesamt, dass die Schleife terminiert.

KORREKTURAUFGABE 54 (7 Punkte) **:**
Gegeben sei das folgende Programm:

$\{n \geq 0\}$
begin
  $i := 0;$
  $x := 0;$
  while $(i < n)$ *do begin*
    $x := x + 2 * i + 1;$
    $i := i + 1;$
  end
end
$\{\ldots\}$

a) Nachfolgend sind 5 Zusicherungen angegeben. Untersuchen Sie, welche davon sich für die Schleife in obigem Programm als Teil einer Invarianten verwenden lassen. Begründen Sie Ihre Entscheidung.
$I_1 = \{i > n\}$, $I_2 = \{i < 0\}$, $I_3 = \{x = i^2\}$, $I_4 = \{x = 2ni\}$, $I_5 = \{i \leq n\}$
(Tipp: Überlegen Sie sich, was das Programm berechnet.)

b) Verifizieren Sie das Programm mit den von Ihnen gefundenen Invarianten. Geben Sie die dazu von Ihnen benutzten Regeln an.

c) Zeigen Sie, dass das Programm terminiert.

Lösung:

a) $I_1 = \{i > n \text{ und } ...\}$ ist zwar invariant für diese Schleife, verhindert jedoch, dass die Schleife überhaupt ausgeführt wird. I_1 kann auch nicht durch Abschwächung aus der Vorbedingung der Schleife erzeugt werden.

$I_2 = \{i < 0 \text{ und } ...\}$ ist keine Invariante, da i in jedem Schleifendurchlauf erhöht wird und diese Eigenschaft somit nicht garantiert werden kann. Außerdem gilt I_2 nicht vor der Schleife.

$I_4 = \{x = 2ni \text{ und } ...\}$ ist ebenfalls keine Invariante, da nach einem Durchlauf durch den Schleifenrumpf $\{x = 2ni - 2n + 2i + 1 \text{ und } ...\}$ gilt. Da i und n ganzzahlig sind, kann $-2n + 2i + 1$ nicht den Wert 0 haben.

$I_3 = \{x = i^2\}$ und $I_5 = \{i \leq n\}$ zusammen eigenen sich als Invariante für die Schleife, wie man in Teil b) sieht.

b)

$\{n \geq 0\}$
begin
$\{n \geq 0\} \rightarrow \{n \geq 0 \text{ und } 0 = 0\}$ — Abschwächungsregel
$i := 0;$ — Zuweisungsregel
$\{n \geq 0 \text{ und } i = 0\} \rightarrow \{n \geq 0 \text{ und } i = 0 \text{ und } 0 = 0\}$ — Abschwächungsregel
$x := 0;$ — Zuweisungsregel
$\{n \geq 0 \text{ und } i = 0 \text{ und } x = 0\}$
$\rightarrow \{n \geq i \text{ und } x = i^2\}$ — Abschwächungsregel
while $(i < n)$ *do begin*
$\{n \geq i \text{ und } x = i^2 \text{ und } i < n\}$ — Schleifenregel
$\rightarrow \{n > i \text{ und } x + 2 * i + 1 = i^2 + 2 * i + 1\}$ — Abschwächungsregel
$x := x + 2 * i + 1;$
$\{n > i \text{ und } x = i^2 + 2 * i + 1\}$ — Zuweisungsregel
$\rightarrow \{n \geq i + 1 \text{ und } x = (i + 1)^2\}$ — Abschwächungsregel
$i := i + 1;$
$\{n \geq i \text{ und } x = i^2\}$ — Zuweisungsregel
end
$\{n \geq i \text{ und } x = i^2 \text{ und } i \geq n\}$ — Schleifenregel
$\rightarrow \{n = i \text{ und } x = i^2\}$ — Abschwächungsregel
$\rightarrow \{x = n^2\}$ — Abschwächungsregel
end
$\{x = n^2\}$

c) Die Variable i ist ganzzahlig, da sie mit dem Wert 0 initialisiert wird und nur durch die Anweisung $i := i + 1;$ verändert wird. i wird in jedem Schleifendurchlauf um 1 erhöht, ist also streng monoton wachsend und ist nach oben durch n beschränkt, da $n \geq i$ Teil der Invariante ist. Daher terminiert die Schleife.

AUFGABE 55 :
Überprüfen Sie, ob die folgenden Programmausschnitte terminieren.

a)
```
while (i > 0) do begin
        x := x − 1;
        if (i > 0) then
            i := x;
    end
```

b)
```
while (c ≥ 0) do begin
        x := c;
        if (x = 0) then
            c := c − x;
        else
            c := c + x;
    end
```

Lösung:

a) In jedem Schleifendurchlauf wird die ganzzahlige (siehe Vorbemerkung zum Übungsblatt) Variable x zunächst um den Wert 1 verringert. Durch die anschließende bedingte Anweisung wird der neue Wert für x der Variablen i zugewiesen, denn da Schleifenbedingung und Bedingung der bedingten Anweisung gleich sind, wird i in jedem Schleifendurchlauf der aktuelle Wert von x zugewiesen.

Die Variable i ist daher ganzzahlig und wird bei jedem Durchlauf durch den Schleifenrumpf um den Wert 1 verringert. Durch die Schleifenbedingung $(i > 0)$ ist i nach unten beschränkt. Also terminiert die Schleife.

b) Die Schleife terminiert für keinen Wert von c mit $c \geq 0$, denn im Fall $c = 0$ wird der Wert von c im Schleifenrumpf nicht verändert und im Fall $c > 0$ wird der Wert von c im Schleifenrumpf vergrößert.

Man kann daher zeigen, dass $\{c \geq 0\}$ invariant ist für die Schleife. Wenn daher nicht $\{c < 0\}$ als Zusicherung vor der Schleife gilt, terminiert die Schleife nicht immer.