Ludwig-Maximilians-Universität München SS 2006
Institut für Informatik Musterlösung Übungsblatt 2
Prof. Dr. M. Wirsing, M. Hammer, A. Rauschmayer

Informatik II Musterlösung

Zu jeder Aufgabe ist eine Datei abzugeben, deren Name rechts in der Aufgabenüberschrift steht. Stellen Sie die insgesamt drei Dateien in ein extra Verzeichnis (mit beliebigem Namen) und packen Sie dieses zu einem ZIP-Archiv. Geben Sie dieses, wie üblich, per UniWorx ab.

Aufgabe 2-1 Programmzustände (`Zustand.java`, 4 Punkte)

Wir stellen den Keller, in dem die Namen und Werte der lokalen Variablen während der Ausführung eines Programms gespeichert werden, durch eine Liste von Name/Wert Paaren dar. Hierbei steht dasjenige Name/Wert Paar am weitesten rechts, das zuletzt zum Keller hinzugefügt wurde.

So ist z.B. $[(x, 12), (y, \textbf{true})]$ eine Darstellung eines Kellers, in dem die Werte der Variablen x und y gespeichert werden. Wird eine neue Variable z, etwa durch `double z = 4.3` vereinbart, ändert sich obiger Keller zu $[(x, 12), (y, \textbf{true}), (z, 4.3)]$.

Betrachten Sie folgendes Java-Programm:

```
1  class Zustand
2  { public static void main(String[] args)
3    { int m = 2;
4      double d = 2.0;
5      d = m * d;
6      int n = 15 + (int) d;
7      { int k = 8;
8        d = d + k;
9      }
10     d = n / 4;
11     System.out.println(d);
12   }
13 }
```

Geben Sie die Zustände des Programms nach Ausführung von Zeile 3, 4, 5, 6, 7, 8 und 10 in der oben beschriebenen Notation an.

Lösung: Wir erhalten folgende Abfolge von Zuständen (links die Nummer der Zeile, nach deren Ausführung der rechts dargestellte Zustand vorliegt):

3 $[(m, 2)]$

4 $[(m, 2), (d, 2.0)]$

5 $[(m, 2), (d, 4.0)]$

6 $[(m, 2), (d, 4.0), (n, 19)]$

7 $[(m, 2), (d, 4.0), (n, 19), (k, 8)]$

8 $[(m, 2), (d, 12.0), (n, 19), (k, 8)]$

10 $[(m, 2), (d, 4.0), (n, 19)]$

Die Variable k taucht in der Beschreibung des letzten Zustands nicht mehr auf, da sie lokal in einem inneren Block deklariert wurde.

Aufgabe 2-2 Quadratzahltest (`Quadratzahl.java`, 4 Punkte)

Eine ganze Zahl $q \geq 0$ ist ein Quadratzahl, wenn es eine ganze Zahl n gibt mit $q = n * n$. So ist z.B. $16 = 4 * 4$ eine Quadratzahl, nicht aber 17.

Vervollständigen Sie das Programmgerüst

```
class Quadratzahl
{   public static void main(String[] args)
    {   int q = 17;
        // Ihr Code
    }
}
```

zu einem Java-Programm, das für jeden gültigen Wert von q ausgibt, ob es sich bei q um eine Quadratzahl handelt.

Hinweis: Prüfen Sie (mit geeigneter Abbruchbedingung), ob $q = n * n$ für $n = 1, 2, \ldots$.

Lösung:

```
class Quadratzahl
{   public static void main(String[] args)
   {   int q = 16;
       int n = 0;
       while (n*n < q) n++;
       if (n*n == q)
           System.out.println("" + q + " ist Quadratzahl.");
       else
           System.out.println("" + q + " ist keine Quadratzahl.");
   }
}
```

Aufgabe 2-3 **Fakultätsfunktion** (`Fakultaet.txt`, 8 Punkte)

Betrachten Sie folgendes Programm-Fragment zur Berechnung der Fakultätsfunktion:

```
public static int fakultaet(int n) {
    // Anfang
    int a = 1;
    int k = n;
    while (k > 0) {
        a = a * k;
        k--;
    }
    // Ende
    return a;
}
```

Sei c der Programmabschnitt zwischen den Kommentaren `Anfang` und `Ende`. Leiten Sie im Hoare-Kalkül das Hoare-Tripel $\{n \geq 0\}\, c\, \{a = n!\}$ – mithin die Korrektheitsaussage für die Methode `fakultaet` – ab. Geben Sie dazu immer die benutzte Regel an sowie die Ableitungen für die Prämissen. Sie können "top-down" oder "bottom-up" vorgehen, also von dem gewünschten Tripel her oder von den ersten Prämissen her (die Sie vermutlich in einem ersten Durchgang auf Papier herausfinden).

Hinweis: Die Schleifen-Invariante enthält unter Anderem die Aussage $a \cdot k! = n!$. Beachten Sie aber den Definitionsbereich von "!".

Lösung: Als Invariante wählen wir

$$I \equiv \mathtt{a} \cdot \mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} \geq 0$$

Das $\mathtt{k} \geq 0$ ist relevant, da die Fakultätsfunktion als $\mathbb{N} \to \mathbb{N}$ definiert ist, jedoch `k` als Integer-Variable auch negative Werte annehmen kann; zudem hilft es für den Nachweis totaler Korrektheit.

$\mathtt{a} \cdot \mathtt{k}! = \mathtt{n}!$ beschreibt den Zusammenhang von `a` und `k`: Nach i Schritten gilt $\mathtt{k} = \mathtt{n} - i$ und $\mathtt{a} = \mathtt{n} \cdot (\mathtt{n}-1) \cdot \ldots \cdot (\mathtt{n} - i + 1)$, und somit ist $\mathtt{a} = \frac{\mathtt{n}!}{\mathtt{k}!}$.

Wir weisen nun zuerst die Invarianz von I nach:

$$\{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} \geq 0\}$$
$$\mathtt{k = k - 1;}$$
$$\{\mathtt{a}\cdot(\mathtt{k}-1)! = \mathtt{n}! \wedge (\mathtt{k}-1) \geq 0\}$$
$$\mathtt{a = a * k;}$$
$$\{\mathtt{a}\cdot\mathtt{k}\cdot(\mathtt{k}-1)! = \mathtt{n}! \wedge (\mathtt{k}-1) \geq 0\}$$

Da $\mathtt{k}\cdot(\mathtt{k}-1)! = \mathtt{k}!$ gilt, können wir dies umschreiben zu

$$\{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge (\mathtt{k}-1) \geq 0\}$$

$\mathtt{k}-1 \geq 0$ ist äquivalent zu $\mathtt{k} > 0$, und dies wiederum äquivalent zu $\mathtt{k} \geq 0 \wedge \mathtt{k} \neq 0$. Damit können wir obrige Vorbedingung erneut umschreiben zu

$$\{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} \geq 0 \wedge \mathtt{k} \neq 0\}$$

Damit erhalten wir das Hoare-Tripel

$$\{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} \geq 0 \wedge \mathtt{k} \neq 0\}\ \mathtt{a = a * k;\ k=k-1;}\ \{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} \geq 0\}$$

Dieser Teil des Beweises, formal geschrieben (aus Platzgründen verwenden wir $\alpha \equiv \mathtt{a}\cdot(\mathtt{k}-1)! = \mathtt{n}! \wedge \mathtt{k}-1 \geq 0$):

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\{\mathtt{a}\cdot\mathtt{k}\cdot(\mathtt{k}-1)! = \mathtt{n}! \wedge \mathtt{k}-1 \geq 0\}\ \mathtt{a = a * k;}\ \{\alpha\}}\ (\text{Zuw}) \quad \dfrac{}{\{\alpha\}\ \mathtt{k = k - 1;}\ \{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} \geq 0\}}\ (\text{Zuw})}{\{\mathtt{a}\cdot\mathtt{k}\cdot(\mathtt{k}-1)! = \mathtt{n}! \wedge \mathtt{k}-1 \geq 0\}\ \mathtt{a = a * k;\ k = k - 1;}\ \{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} \geq 0\}}\ (\text{Seq})}{\{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge \mathtt{k}-1 \geq 0\}\ \mathtt{a = a * k;\ k = k - 1;}\ \{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} \geq 0\}}\ (\text{Abschw})}{\{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} \geq 0 \wedge \mathtt{k} \neq 0\}\ \mathtt{a = a * k;\ k = k - 1;}\ \{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} \geq 0\}}\ (\text{Abschw})$$

Damit ist die Invariante gezeigt. Wir wenden noch die Blockregel an und erhalten

$$\mathbf{F_{part}} \equiv \{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} \geq 0 \wedge \mathtt{k} \neq 0\}\ \mathtt{\{a = a * k;\ k = k - 1;\}}\ \{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} \geq 0\}$$

Für die totale Korrektheit setzen wir $t = \mathtt{k}$. Es gilt $I \implies \mathtt{k} \geq 0$. Um zu zeigen, daß $\mathtt{k}$ bei jedem Schritt kleiner wird, verwenden wir den „Trick“, den Wert, um den $\mathtt{k}$ tatsächlich kleiner wird, konkret anzugeben und dann zu einem unquantifizierenden „kleiner“ abzuschwächen:

$$\{\mathtt{k} = z - 1\}$$
$$\mathtt{k = k - 1;}$$
$$\{\mathtt{k} - 1 = z - 1\}$$
$$\mathtt{a = a * k;}$$
$$\{\mathtt{k} - 1 = z - 1\}$$

Damit können wir wie folgt fortfahren:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\{\mathtt{k}-1 = z-1\}\ \mathtt{a = a * k;\ k = k - 1;}\ \{\mathtt{k} = z-1\}}\ (\text{Abschw})}{\{\mathtt{k} = z\}\ \mathtt{a = a * k;\ k = k - 1;}\ \{\mathtt{k} < z\}}\ (\text{Abschw})}{\{I \wedge \mathtt{k} \geq 0 \wedge \mathtt{k} = z\}\ \mathtt{a = a * k;\ k = k - 1;}\ \{\mathtt{k} < z\}}\ (\text{Block})}{\{I \wedge \mathtt{k} \geq 0 \wedge \mathtt{k} = z\}\ \mathtt{\{a = a * k;\ k = k - 1;\ \}}\ \{\mathtt{k} < z\}} \quad \mathbf{F_{part}}}{\{I\}\ \mathtt{while\ (k > 0)\ \{a = a * k;\ k = k - 1;\ \}}\ \{I \wedge \neg(\mathtt{k} > 0)\}}\ (\text{Iteration}_{\text{total}})}{\{I\}\ \mathtt{while\ (k > 0)\ \{a = a * k;\ k = k - 1;\ \}}\ \{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} \geq 0 \wedge \mathtt{k} \leq 0\}}\ (\text{Abschw})}{\{I\}\ \mathtt{while\ (k > 0)\ \{a = a * k;\ k = k - 1;\ \}}\ \{\mathtt{a}\cdot\mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} = 0\}}\ (\text{Abschw})}{\{I\}\ \mathtt{while\ (k > 0)\ \{a = a * k;\ k = k - 1;\ \}}\ \{\mathtt{a} = \mathtt{n}!\}}\ (\text{Abschw})$$

(Beachte bei der letzten Anwendung der Abschwächungsregel, dass $0! = 1$ gilt!)
Jetzt bleibt nur noch zu zeigen, daß die Invariante aus dem Initialisierungsteil folgt:

$$\begin{array}{c}\{\mathtt{a} \cdot \mathtt{k}! = \mathtt{n}! \wedge \mathtt{k} \geq 0\} \\ \texttt{k = n;} \\ \{\mathtt{a} \cdot \mathtt{n}! = \mathtt{n}! \wedge \mathtt{n} \geq 0\} \\ \texttt{a = 1;} \\ \{1 \cdot \mathtt{n}! = \mathtt{n}! \wedge \mathtt{n} \geq 0\}\end{array}$$

Und das liefert den Gesamtbeweis:

$$\dfrac{\dfrac{\{1 \cdot \mathtt{n}! = \mathtt{n}! \wedge \mathtt{n} \geq 0\}\ \texttt{a = 1; k = n;}\ \{I\}}{\{\mathtt{n} \geq 0\}\ \texttt{a = 1; k = n;}\ \{I\}}\ (\text{Abs.}) \qquad \{I\}\ \texttt{while (k > 0) \{a = a * k; k = k - 1; \}}\ \{\mathtt{a} = \mathtt{n}!\}}{\{\mathtt{n} \geq 0\}\ c\ \{\mathtt{a} = \mathtt{n}!\}}\ (\text{Seq})$$

Abgabe: Per UniWorx, bis spätestens 15.5.2006, 9:00 Uhr.