



Abgabe: 18.11.2008 (vor der Vorlesung)

### Aufgabe 5.1 (H) Tripel

a) Gegeben sei folgende MiniJava-Prozedur:

```
void f() {  
    z = x;  
    x = y;  
    y = z;  
}
```

Welche der folgenden Tripel sind gültig. Begründen Sie Ihre Antworten.

- i)  $\{x = l_x \wedge y = l_y \wedge l_x \leq l_y\} \quad f(); \quad \{x = l_y \wedge y = l_x \wedge l_y \leq l_x\}$
- ii)  $\{x = l_x \wedge y = l_y \wedge l_x \leq l_y\} \quad f(); \quad \{x = l_y \wedge y = l_x \wedge l_x \leq l_y\}$
- iii)  $\{x = l_x \wedge y = l_y \wedge l_x \leq l_y\} \quad f(); \quad \{x = l_x \wedge y = l_y \wedge l_y \leq l_x\}$

b) Geben Sie eine Prozedur  $f()$  an, für die das Tripel

$$\{\text{false}\} \quad f(); \quad \{\text{true}\}$$

gültig ist. Begründen Sie Ihre Antwort.

c) Im folgenden seien  $x, y, z, n$  globale Programmvariablen. Nehmen Sie an, dass folgendes Tripel für die nicht näher spezifizierte Prozedur  $f()$  gültig sei:

$$\{x = l_x \wedge y = l_y \wedge |l_x| \leq 2^{l_y} \wedge l_y \geq 0\} \quad f(); \quad \{x = 0 \wedge z = l_x \cdot l_y + 42\} \quad (1)$$

Welche der folgenden Tripel sind unter dieser Annahme gültig. Begründen Sie Ihre Antworten.

Für ein Tripel dessen Gültigkeit nicht durch die Gültigkeit des Tripels in (1) impliziert wird, ist ein Gegenbeispiel anzugeben. D.h. es ist eine Prozedur  $f()$  anzugeben für die das Tripel in (1) erfüllt ist, aber das Tripel in der entsprechenden Teilaufgabe nicht erfüllt ist. Um zu zeigen, dass ein Tripel nicht gilt sind lediglich Werte für  $x$  und  $y$  zu benennen, für die das Tripel nicht gilt.

- i)  $\{x = l_x \wedge y = l_y \wedge |l_x| \leq 2^{l_y} \wedge l_y \geq 0\} \quad f(); \quad \{y = l_y\}$
- ii)  $\{x = l_x + 2 \cdot l_y \wedge y = l_y \wedge |l_x + 2 \cdot l_y| \leq 2^{l_y} \wedge l_y \geq 0\} \quad f(); \quad \{x = 0 \wedge z = l_x \cdot l_y + 2 \cdot l_y \cdot l_y + 42\}$
- iii)  $\{x = l_x \wedge y = l_y \wedge |l_x| \leq 2^{l_y} \wedge l_y \geq 0 \wedge n \leq 5\} \quad f(); \quad \{x = 0 \wedge z = l_x \cdot l_y + 42 \wedge n \leq 5\}$
- iv)  $\{x = l_x \wedge y = 2 \cdot l_y \wedge |l_x| = 4^{l_y} \wedge 2 \cdot l_y \geq 0\} \quad f(); \quad \{x \leq 0 \wedge z = 2 \cdot l_x \cdot l_y + 42\}$

**Aufgabe 5.2 (H)**

Gegeben sei folgende MiniJava-Prozedur:

```
void f() {
  while (y > 0) {
    x = x * x;
    y = y - 1;
  }
}
```

Geben Sie ein möglichst aussagekräftiges gültiges Tripel  $\{A\} f(); \{B\}$  für die Prozedur  $f()$  an und zeigen Sie dessen Gültigkeit.

**Aufgabe 5.3 (P) Rekursive Prozeduren**

Gegeben sei folgendes MiniJava-Programm:

```
int n, m, r;

void f() {
  if (n >= m && m >= 0) {
    if (m == 0)
      r = r + 1;
    else {
      n = n - 1;
      m = m - 1;
      f();
      m = m + 1;
      f();
      n = n + 1;
    }
  }
}
```

- Erstellen Sie das Kontrollfluss-Diagramm für die Prozedur  $f()$ !
- Zeigen Sie die Gültigkeit des Tripels

$$\{r = l_r \wedge m = l_m \wedge n = l_n\} f(); \quad \{r = l_r + \binom{l_n}{l_m} \wedge m = l_m \wedge n = l_n\}.$$

Als Hilfestellung sei an die folgende Identität erinnert:

$$\binom{n}{m} = \begin{cases} 1 & \text{falls } m = 0, n \geq 0 \\ \binom{n-1}{m-1} + \binom{n-1}{m} & \text{falls } n \geq m \geq 1 \\ 0 & \text{sonst} \end{cases}, \quad m, n \in \mathbb{Z}.$$

### Aufgabe 5.4 (P) Was ist gleich?

Alternative a	Alternative b	gleich	nicht gleich
<code>int -&gt; (int -&gt; int)</code>	<code>(int -&gt; int) -&gt; int</code>	<input type="checkbox"/>	<input type="checkbox"/>
<code>(int -&gt; int) -&gt; int</code>	<code>int -&gt; int -&gt; int</code>	<input type="checkbox"/>	<input type="checkbox"/>
<code>int -&gt; (int -&gt; int)</code>	<code>int -&gt; int -&gt; int</code>	<input type="checkbox"/>	<input type="checkbox"/>
<code>a b c</code>	<code>(a b) c</code>	<input type="checkbox"/>	<input type="checkbox"/>
<code>a b c</code>	<code>a (b c)</code>	<input type="checkbox"/>	<input type="checkbox"/>
<code>x</code>	<code>((x))</code>	<input type="checkbox"/>	<input type="checkbox"/>
<code>x, y</code>	<code>(x, y)</code>	<input type="checkbox"/>	<input type="checkbox"/>
<code>1::2::3::[]</code>	<code>1::[2;3]::[]</code>	<input type="checkbox"/>	<input type="checkbox"/>
<code>1::2::3::[]</code>	<code>1::[2;3]</code>	<input type="checkbox"/>	<input type="checkbox"/>

### Aufgabe 5.5 (P) OCaml vs. Java!

Übersetzen Sie die statischen Methoden der folgenden Java-Klasse möglichst genau in OCaml.

```

public class Java {
    public static int sign(int a) {
        if (a > 0)
            return 1;
        else if (a < 0)
            return -1;
        return 0;
    }

    public static int sum_n(int n) {
        if (n == 0)
            return 0;
        return n + sum_n(n - 1);
    }

    public static int sum_n_2(int n) {
        int erg = 0;
        while (n > 0) {
            erg = erg + n;
            n = n - 1;
        }
        return erg;
    }
}

```

### Aufgabe 5.6 (P) Finanzgeschäfte

In dieser Aufgabe soll die Situation in einer Bank betrachtet werden.

- a) Definieren Sie einen Datentyp `wertpapier` zur Repräsentation von Wertpapieren, wie z.B. Aktien oder Zertifikate. Ein Wertpapier hat einen Namen und einen Wert.
- b) Des Weiteren wird ein Datentyp `kunde` zur Repräsentation von Kunden benötigt. Jeder Kunde hat einen Namen und ein Portfolio, welches eine Liste von, aus einer Anzahl und einem Wertpapier bestehenden, Paaren ist.
- c) Schreiben Sie eine Funktion, die den Gesamtwert eines Portfolios bestimmt.
- d) Schreiben Sie eine Funktion, die das in Wertpapieren angelegte „Vermögen“ eines Kunden ermittelt.
- e) Schreiben Sie eine Funktion `anzahl : (kunde -> bool) -> kunde list -> int`, für die der Aufruf `anzahl p l` die Anzahl der Kunden in der Liste `l`, die das Prädikat `p` erfüllen, ergibt.
- f) Schreiben Sie eine Funktion `filter : (kunde -> bool) -> kunde list -> kunde list`, für die der Aufruf `filter p l` die Liste der Kunden aus der Liste `l`, die das Prädikat `p` erfüllen, ergibt.
- g) Schreiben Sie eine Funktion `grosskunden : kunde list -> kunde list`, für die der Aufruf `grosskunden l` die Liste derjenigen Kunden aus der Liste `l`, deren Portfolio mindestens einen Gesamtwert von 1000000 \$ hat, ergibt.