



Abgabe: 11.11.2008 (vor der Vorlesung)

Aufgabe 4.1 (H) Terminierung

Zeigen Sie, dass folgendes MiniJava-Programm stets terminiert.

```
int i, k, n;  
n = read();  
if (n < 0)  
    n = -1 * n;  
x = 1;  
i = 0;  
while (i != n && i != n+1) {  
    x = 2 * x;  
    i = i + 2;  
}
```

Aufgabe 4.2 (H) Binomialkoeffizient

Gegeben sei folgendes Code-Fragment:

```
int i, k, n, b;  
  
n = read();  
k = read();  
if (0 <= k && k <= n) {  
    b = 1;  
    i = 1;  
    while (i <= k) {  
        b = b * (n + 1 - i);  
        b = b / i;  
        i = i + 1;  
    }  
} else {  
    b = 0;  
}  
write(b);
```

Zeigen Sie, dass das gegebene Code-Fragment den Binomialkoeffizienten $\binom{n}{k}$ berechnet und ausgibt. Dieser ist durch

$$\binom{n}{k} = \begin{cases} \frac{n!}{(n-k)! \cdot k!} & \text{falls } 0 \leq k \leq n \text{ gilt} \\ 0 & \text{andernfalls} \end{cases}, \quad n, k \in \mathbb{Z}$$

gegeben. Am Stop-Knoten muss also die Zusicherung $b = \binom{n}{k}$ gelten.

Hinweis: Ein Korrektheitsbeweis sichert auch, dass, unmittelbar vor der Ausführung der Anweisung $b = b/i$, b immer durch i teilbar ist, d.h. es gilt $\frac{b}{i} \in \mathbb{Z}$ unmittelbar vor der Anweisung.

Aufgabe 4.3 (P) Adaption

Nehmen Sie an, dass folgendes Tripel für die nicht näher spezifizierte Prozedur $f()$ gültig sei:

$$\{x = l_x \wedge y = l_y \wedge l_x \leq l_y\} \quad f(); \quad \{z = 12 \cdot l_x - 3 \cdot l_y \wedge y = l_y\} \quad (1)$$

- Wie ist der Wert der Variablen z nach Ausführung der Prozedur $f()$, wenn vor Ausführung x den Wert 1 und y den Wert 0 hat?
- Wie ist der Wert der Variablen z nach Ausführung der Prozedur $f()$, wenn vor Ausführung x den Wert 1 und y den Wert 2 hat?
- Wie ist der Wert der Variablen x nach Ausführung der Prozedur $f()$, wenn vor Ausführung x den Wert 7 hat?

- Welche der folgenden Tripel sind gültig? Begründen Sie Ihre Antwort!

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 aber das Tripel in der 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.

- $\{x = l_x = y = l_y\} \quad f(); \quad \{z = 12 \cdot l_x - 3 \cdot l_y \wedge y = l_y\}$
 - $\{x = l_x \wedge y = l_y\} \quad f(); \quad \{z = 12 \cdot l_x - 3 \cdot l_y \wedge y = l_y\}$
 - $\{x = l_x \wedge y = l_y \wedge l_x \leq l_y\} \quad f(); \quad \{x = l_x \Rightarrow (z = 12 \cdot l_x - 3 \cdot l_y \wedge y = l_y)\}$
 - $\{x = l_x \wedge y = l_y + l_x + 1 \wedge -1 \leq l_y\} \quad f(); \quad \{z = 9 \cdot l_x - 3 \cdot l_y - 3 \wedge y = l_y + l_x + 1\}$
 - $\{0 < x \wedge x < y \wedge y < 0\} \quad f(); \quad \{z = 42\}$
- Geben Sie eine Prozedur $f()$ an, für die das Tripel

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

erfüllt ist.

Aufgabe 4.4 (P) Prozeduren

Gegeben sei folgendes Code-Fragment:

```
int n, r;

public void fact() {
    /* Ihr Code */
}
```

- Ergänzen Sie das Fragment zu einer **nicht-rekursiven** Prozedur für die das Tripel

$$\{n = l \geq 0\} \quad \text{fact}(); \quad \{r = l!\}$$

gilt, wobei l eine logische Variable ist. Zur Erinnerung:

$$l! = \prod_{i=1}^l i = l \cdot (l-1) \cdot (l-2) \cdot \dots \cdot 1, \quad l \geq 0.$$

Im Speziellen gilt: $0! = 1$.

- Zeigen Sie die Korrektheit Ihrer Implementierung!