8. Übungsblatt zu Logik, Semantik und Verifikation SS 2002

Prof. Dr. Gert Smolka, Dipl.-Inform. Tim Priesnitz
www.ps.uni-sb.de/courses/prog-lsv02/

Abgabe: Montag, 3. Juni in der Vorlesungspause

Aufgabe 8.1: Grundregeln (5) Sei die Menge $Q \subseteq \mathbb{N} \times \mathbb{N}$ durch die folgenden Inferenzregeln definiert:

$$\frac{}{\langle 1, 1\rangle \in Q} \qquad \frac{\langle n, m\rangle \in Q \qquad n' = n + 1 \qquad m' = m + 2n + 1}{\langle n', m'\rangle \in Q}$$

(a) Geben Sie die durch die Inferenzregeln definierte Grundregelmenge R an.

(b) Geben Sie die Mengen $\hat{R}^0(\emptyset)$, $\hat{R}^1(\emptyset)$, $\hat{R}^2(\emptyset)$, $\hat{R}^3(\emptyset)$ und $\hat{R}^4(\emptyset)$ an.

Aufgabe 8.2: Grundregeln und Fixpunkte (8) Sei die Menge $M \subseteq \mathbb{Z}$ durch die folgenden Inferenzregeln definiert:

$$\frac{}{4 \in M} \qquad \frac{x \in M \qquad y \in M \qquad z = (x + y) \bmod 5}{z \in M}$$

(a) Geben Sie die durch die Inferenzregeln definierte Grundregelmenge R an.

(b) Geben Sie die Mengen $\hat{R}^0(\emptyset)$, $\hat{R}^1(\emptyset)$, $\hat{R}^2(\emptyset)$, $\hat{R}^3(\emptyset)$ und $\hat{R}^4(\emptyset)$ an.

(c) Geben Sie die Menge $\bigcup_{i \in \mathbb{N}} \hat{R}^i(\emptyset)$ an.

(d) Geben Sie den kleinsten Fixpunkt von $\hat{R}$ an.

(e) Geben Sie das kleinste $P \subseteq \mathbb{Z}$ an mit $\hat{R}(P) \subseteq P$.

Aufgabe 8.3: Regeln mit unendlich vielen Prämissen (8) Sei X eine Menge und R eine beliebige Teilmenge von $\mathcal{P}(X) \times X$. Wir definieren

$$\hat{R} \in \mathcal{P}(X) \to \mathcal{P}(X)$$

$$\hat{R}(A) = \{x \mid \exists B : \; B \subseteq A \text{ und } \langle B, x\rangle \in R\}$$

Geben Sie eine einelementige Menge $R \subseteq \mathcal{P}(\mathbb{N}) \times \mathbb{N}$ an, so dass $\hat{R}$ nicht stetig ist (bezüglich der VPO $\langle \mathcal{P}(\mathbb{N}), \subseteq\rangle$). Beweisen Sie diese Aussage.

Aufgabe 8.4: Unstetige Funktionen (8) Geben Sie eine Funktion $F \in (\mathbb{N} \to \mathbb{N}_\perp) \to (\mathbb{N} \to \mathbb{N}_\perp)$ an, die monoton aber nicht stetig ist.

Aufgabe 8.5: Approximation von Funktionen (6) Gegeben sei die folgende rekursive Prozedurdeklaration:

```
fun max(n,m) = if n=0 then m
               else if m=0 then n
                    else 1 + max(n-1,m-1)
val max : int * int -> int
```

(a) Geben Sie das der Deklaration entsprechende Funktional `maxFun` an.

(b) Geben Sie die ersten vier Approximationen für `max` an.

(c) Geben Sie eine Prozedur `approx: int -> (int*int) -> int` an, die zu $k \in \mathbb{N}$ die k-te Appriximation für `max` liefert.

Aufgabe 8.6: Denotationale Semantik für IMP (10) Sie sollen die denotationale Semantik von IMP in Standard ML implementieren. Dabei sollen Sie die folgenden Deklarationen verwenden:

```
type con = int
type loc = string

datatype aexp = Con of con | Loc of loc | Add of aexp * aexp

datatype bexp = LE of aexp * aexp

datatype com = Assign of loc * aexp
             | Seq    of com * com
             | If     of bexp * com * com
             | While  of bexp * com

type sigma = loc -> int

fun fix f x = f (fix f) x
```

Gehen Sie wie folgt vor:

(a) Implementieren Sie die denotationale Semantik für arithmetische Ausdrücke:

```
val da : aexp -> sigma -> int
```

(b) Implementieren Sie die denotationale Semantik für Boolesche Ausdrücke:

```
val db : bexp -> sigma -> bool
```

(c) Implementieren Sie die Hilfsfunktion für Schleifen:

```
val gamma : (sigma -> bool) * (sigma -> sigma) ->
            (sigma -> sigma) -> (sigma -> sigma)
```

(d) Implementieren Sie die denotationale Semantik für Kommandos:

```
val dc : com -> sigma -> sigma
```

Aufgabe 8.7: Programmieren in IMP (5) Sei die Prozedur

```
fun eval c n = dc c (fn l => if l="argument" then n else 0) "result"
val eval : com -> int -> int
```

gegeben, die zu einem Kommando eine Prozedur `int->int` liefert, die das Kommando mit dem in der Lokation `argument` übergebenen Argument ausführt und nach der Terminierung den Wert der Lokation `result` liefert.
Geben Sie ein Kommando *c* an, für das eval eine Prozedur liefert, die die Fakultätsfunktion berechnet. Denken Sie daran, dass IMP keine Multiplikation besitzt.