Musterlösung zum 2. Übungsblatt "Informatik I" WS 95/96

Frank Berwanger, Stephan Bergmann, Georg Demme, Christian Schulte

Aufgabe II.1 (Namen eines Ausdrucks)
$\mathrm{N}[N] = N$
$\mathrm{N}[K] = \emptyset$
$\mathrm{N}[($`if` $A_1\ A_2\ A_3)] = \mathrm{N}[A_1] \cup \mathrm{N}[A_2] \cup \mathrm{N}[A_3]$
$\mathrm{N}[(A_1\ ...\ A_n)] = \mathrm{N}[A_1]\ \cup\ ...\ \cup\ \mathrm{N}[A_n]$
$\mathrm{N}[N[($`lambda` $(N_1\ ...\ N_k)\ A)]\ =\ \{N_1,\ ...,\ N_k\} \cup \mathrm{N}[A]$

Aufgabe II.2 (Iterative Exponentialfunktion) Induktive Definition :

$$\exp(n, m, k) = \begin{cases} k & \text{falls } m = 0 \\ \exp(n, m-1, k*n) & \text{sonst} \end{cases}$$

Daraus ergibt sich folgende Scheme-Funktion:

```
(define (exp n m k)
    (if (< m 1)
          k
          (exp n (- m 1) (* m k))))
```

Aufgabe II.3 (Wieso ist das Konditional primitiv? (4)) Da beim Aufruf von `konditional` **immer** alle drei Argumente evaluiert werden, führt der Aufruf von `fak n` zwangsläufig zum Aufruf von `fak (- n 1)`, was zu einer unendlichen Rekursion führt. Selbst wenn `n` $= 0$ ist, wird `fak (- 0 1)` evaluiert, also `fak (-1)`.
Dies geschieht nicht, wenn man `if` benutzt. Hierbei wird zuerst die Bedingung evaluiert und erst dann in Abhängigkeit des Ergebnisses einer der beiden anderen Ausdrücke.

Aufgabe II.4 (Semantikbäume) Sei G die globale Umgebung, die nur die primitiven Prozeduren enthält.

1.

`(define x (if (if (= 2 3) (= 0 0) (= 2 3)) (+ 1 2) (+ 3 4)))`$, G \to G[7/\mathrm{x}]$

- `(if (if (= 2 3) (= 0 0) (= 2 3)) (+ 1 2) (+ 3 4))`$, \emptyset, G \to 7$
  - `(if (= 2 3) (= 0 0) (= 2 3))`$, \emptyset, G \to$ `#f`
    - `(= 2 3)`$, \emptyset, G \to$ `#f`
      - `=`$, \emptyset, G \to \boxed{=}$
      - `2`$, \emptyset, G \to 2$
      - `3`$, \emptyset, G \to 3$
    - `(= 2 3)`$, \emptyset, G \to$ `#f`
      - `=`$, \emptyset, G \to \boxed{=}$
      - `2`$, \emptyset, G \to 2$
      - `3`$, \emptyset, G \to 3$
  - `(+ 3 4)`$, \emptyset, G \to 7$
    - `+`$, \emptyset, G \to \boxed{+}$
    - `3`$, \emptyset, G \to 3$
    - `4`$, \emptyset, G \to 4$

2.

$\underline{\begin{array}{l}\texttt{(define quadrat (lambda (x) (* x x)))}, G[7/\texttt{x}] \\ \quad \rightarrow G[7/\texttt{x}, \texttt{(lambda (x) (* x x))}\Box\emptyset/\texttt{quadrat}]\end{array}}$

- $\texttt{(lambda (x) (* x x))}, \emptyset, G[7/\texttt{x}] \rightarrow \texttt{(lambda (x) (* x x))}\Box\emptyset$

3. Sei $G' := G[7/\texttt{x}, \texttt{(lambda (x) (* x x))}\Box\emptyset/\texttt{quadrat}]$.

$\underline{\texttt{((lambda (x y) (+ (x) (quadrat y))) (lambda () x) 3)}, \emptyset, G' \rightarrow 16}$

- $\texttt{(lambda (x y) (+ (x) (quadrat y)))}, \emptyset, G'$
  $\rightarrow \texttt{(lambda (x y) (+ (x) (quadrat y)))}\Box\emptyset$
- $\texttt{(lambda () x)}, \emptyset, G' \rightarrow \texttt{(lambda () x)}\Box\emptyset$
- $\texttt{3}, \emptyset, G' \rightarrow 3$
- $\underline{\texttt{(+ (x) (quadrat y))}, \emptyset[\texttt{(lambda () x)}\Box\emptyset/\texttt{x}, 3/\texttt{y}], G' \rightarrow 16}$
  - $\texttt{+}, \emptyset[\texttt{(lambda () x)}\Box\emptyset/\texttt{x}, 3/\texttt{y}], G' \rightarrow \boxed{+}$
  - $\underline{\texttt{(x)}, \emptyset[\texttt{(lambda () x)}\Box\emptyset/\texttt{x}, 3/\texttt{y}], G' \rightarrow 7}$
    - $\texttt{x}, \emptyset[\texttt{(lambda () x)}\Box\emptyset/\texttt{x}, 3/\texttt{y}], G' \rightarrow \texttt{(lambda () x)}\Box\emptyset$
    - $\texttt{x}, \emptyset, G' \rightarrow 7$
  - $\underline{\texttt{(quadrat y)}, \emptyset[\texttt{(lambda () x)}\Box\emptyset/\texttt{x}, 3/\texttt{y}], G' \rightarrow 9}$
    - $\texttt{quadrat}, \emptyset[\texttt{(lambda () x)}\Box\emptyset/\texttt{x}, 3/\texttt{y}], G' \rightarrow \texttt{(lambda (x) (* x x))}\Box\emptyset$
    - $\texttt{y}, \emptyset[\texttt{(lambda () x)}\Box\emptyset/\texttt{x}, 3/\texttt{y}], G' \rightarrow 3$
    - $\underline{\texttt{(* x x)}, \emptyset[3/\texttt{x}], G' \rightarrow 9}$
      - $\texttt{*}, \emptyset[3/\texttt{x}], G' \rightarrow \boxed{*}$
      - $\texttt{x}, \emptyset[3/\texttt{x}], G' \rightarrow 3$
      - $\texttt{x}, \emptyset[3/\texttt{x}], G' \rightarrow 3$

4.

$\underline{\begin{array}{l}\texttt{(define fak (lambda (n) (if (= n 0) 1 (* n (fak (- n 1))))))}, \emptyset, G' \\ \quad \rightarrow G'[\texttt{(lambda (n) (if (= n 0) 1 (* n (fak (- n 1)))))}\Box\emptyset/\texttt{fak}]\end{array}}$

- $\texttt{(lambda (n) (if (= n 0) 1 (* n (fak (- n 1)))))}, \emptyset, G'$
  $\rightarrow \texttt{(lambda (n) (if (= n 0) 1 (* n (fak (- n 1)))))}\Box\emptyset$

5. Sei

$$\begin{array}{ll} G'' := G[7/\texttt{x}, & \\ & \texttt{(lambda (x) (* x x))}\Box\emptyset/\texttt{quadrat}, \\ & \texttt{(lambda (n) (if (= n 0) 1 (* n (fak (- n 1)))))}\Box\emptyset/\texttt{fak}] \end{array}$$

Aufgabe II.5 (Kubikwurzel)

```
(define (qroot x y z)
   (if (< abs(- x (* y y y)) z)
         y
         (qroot x (/ (+ (* 2 y) (/ x (* y y))) 3) z)))

(define (kubikwurzel x) (qroot x 1.0 0.001))
```

Aufgabe II.6 (Abgeleitete Formen)

1. ```
   (define null? (lambda (n)
             (= n 0)))
   ```

2. ```
   (define kleiner (lambda (n m)
             (if (= n 0)
                 (if (= m 0)
                     #t
                     #f)
                 (if (= m 0)
                     #f
                     (kleiner (- n 1) (- m 1))))))
   ```

Zusatzaufgabe II.7 (Semantikbäume)

```
((lambda (x) (x x)) (lambda (x) (x x)))
```

hat keinen Semantikbaum, weil das evaluieren des Ausdruckes nie terminiert!