

# Frei vorkommende Namen und Substitution

Die Menge  $free(e)$  aller in  $e$  frei vorkommenden Namen ist definiert durch

$$\begin{aligned} free(c) &= \emptyset \\ free(id) &= \{id\} \\ free(e_1 e_2) &= free(e_1) \cup free(e_2) \\ free(\mathbf{if} e_0 \mathbf{then} e_1 \mathbf{else} e_2) &= free(e_0) \cup free(e_1) \cup free(e_2) \\ free(\lambda id. e) &= free(e) \setminus \{id\} \end{aligned}$$

$e$  heißt *abgeschlossen*, wenn  $free(e) = \emptyset$ .

**Ergänzung**

Der Ausdruck  $e'[e/id]$ , der durch *Substitution von  $id$  durch  $e$*  aus  $e'$  entsteht, ist definiert durch

$$\begin{aligned} c[e/id] &= c \\ id'[e/id] &= \begin{cases} e & \text{falls } id = id' \\ id' & \text{sonst} \end{cases} \\ (e_1 e_2)[e/id] &= (e_1[e/id])(e_2[e/id]) \\ (\mathbf{if} e_0 \mathbf{then} e_1 \mathbf{else} e_2)[e/id] &= \mathbf{if} e_0[e/id] \mathbf{then} e_1[e/id] \mathbf{else} e_2[e/id] \\ (\lambda id'. e_1)[e/id] &= \begin{cases} \lambda id'. e_1 & \text{falls } id' = id \\ \lambda id''. e_1[id''/id'][e/id] & \text{sonst} \\ \text{mit 'neuem' Namen } id'' & \end{cases} \end{aligned}$$

**Korrektur**

Im letzten Fall bedeutet 'neu', dass  $id'' \notin free(\lambda id'. e_1) \cup free(e) \cup \{id\}$ . Deshalb kann man  $id'' = id'$  wählen, wenn  $id' \notin free(e)$ , also insbesondere wenn  $e$  abgeschlossen ist. Damit vereinfacht sich die Gleichung in diesem Fall zu

$$(\lambda id'. e_1)[e/id] = \lambda id'. (e_1[e/id])$$

## Small step Semantik von $\mathcal{L}_1$ (und $\mathcal{L}_0$ )

Vorgegeben seien

- eine Menge  $Exn$  von *Ausnahmen* (engl.: *exceptions*)  $exn$
- für jeden arithmetischen Operator  $op$  eine Funktion

$$op^{\mathcal{I}} : Int \times Int \rightarrow Int \cup Exn$$

- für jeden Vergleichsoperator  $op$  eine Funktion

$$op^{\mathcal{I}} : Int \times Int \rightarrow Bool$$

Die Menge  $Val \subseteq Exp$  der Werte (engl.: *values*)  $v$  wird definiert durch

$$\begin{array}{l}
 v ::= c \\
 \quad | \textit{id} \\
 \quad | \textit{op } n \\
 \quad | \lambda \textit{id}. e
 \end{array}$$

Ein *small step* ist eine ‘Formel’ der Gestalt  $e \rightarrow e'$  oder  $e \rightarrow \textit{exn}$ . Ein small step heißt *gültig*, wenn er sich mit den folgenden Regeln herleiten lässt:

$$\begin{array}{ll}
 \text{(OP)} & \textit{op } n_1 n_2 \rightarrow \textit{op}^{\mathcal{I}}(n_1, n_2) \\
 \text{(BETA-V)} & (\lambda \textit{id}. e) v \rightarrow e[v/\textit{id}] \\
 \\
 \text{(APP-LEFT)} & \frac{e_1 \rightarrow e'_1}{e_1 e_2 \rightarrow e'_1 e_2} \\
 \text{(APP-RIGHT)} & \frac{e \rightarrow e'}{v e \rightarrow v e'} \\
 \\
 \text{(COND-EVAL)} & \frac{e_0 \rightarrow e'_0}{\mathbf{if } e_0 \mathbf{ then } e_1 \mathbf{ else } e_2 \rightarrow \mathbf{if } e'_0 \mathbf{ then } e_1 \mathbf{ else } e_2} \\
 \text{(COND-TRUE)} & \mathbf{if } \textit{true} \mathbf{ then } e_1 \mathbf{ else } e_2 \rightarrow e_1 \\
 \text{(COND-FALSE)} & \mathbf{if } \textit{false} \mathbf{ then } e_1 \mathbf{ else } e_2 \rightarrow e_2 \\
 \\
 \text{(LET-EVAL)} & \frac{e_1 \rightarrow e'_1}{\mathbf{let } \textit{id} = e_1 \mathbf{ in } e_2 \rightarrow \mathbf{let } \textit{id} = e'_1 \mathbf{ in } e_2} \\
 \text{(LET-EXEC)} & \mathbf{let } \textit{id} = v \mathbf{ in } e \rightarrow e[v/\textit{id}] \\
 \\
 \text{(APP-LEFT-EXN)} & \frac{e_1 \rightarrow \textit{exn}}{e_1 e_2 \rightarrow \textit{exn}} \\
 \text{(APP-RIGHT-EXN)} & \frac{e \rightarrow \textit{exn}}{v e \rightarrow \textit{exn}} \\
 \text{(COND-EVAL-EXN)} & \frac{e_0 \rightarrow \textit{exn}}{\mathbf{if } e_0 \mathbf{ then } e_1 \mathbf{ else } e_2 \rightarrow \textit{exn}} \\
 \text{(LET-EVAL-EXN)} & \frac{e_1 \rightarrow \textit{exn}}{\mathbf{let } \textit{id} = e_1 \mathbf{ in } e_2 \rightarrow \textit{exn}}
 \end{array}$$

Die letzten vier Regeln dienen nur zum ‘Weiterreichen’ von exceptions. Man erhält sie aus den übrigen small step Regeln durch folgende ‘Meta-Regel’:

Zu jeder Regel der Form (R)  $\frac{e_1 \rightarrow e'_1}{e_2 \rightarrow e'_2}$  (d.h. zu jeder Regel mit Prämisse)

nehme man die Regel (R-EXN)  $\frac{e_1 \rightarrow \textit{exn}}{e_2 \rightarrow \textit{exn}}$  hinzu.