



Auswertungs- und Äquivalenzregeln der BSL

Auswertungsregeln

(KONG) $E[e_1] \rightarrow E[e_2]$ falls $e_1 \rightarrow e_2$
 Skript 8.6

$\langle E \rangle ::= []$
 $| (\langle name \rangle \langle v \rangle^* \langle E \rangle \langle e \rangle^*)$
 $| (\text{cond} [\langle E \rangle \langle e \rangle] \{ [\langle e \rangle \langle e \rangle] \}^*)$

(PROG) Ein Programm wird von links nach rechts ausgeführt und startet mit der leeren Umgebung. Ist das nächste Programmelement eine Funktions- oder Strukturdefinition, so wird diese Definition in die Umgebung aufgenommen und die Ausführung mit dem nächsten Programmelement in der erweiterten Umgebung fortgesetzt. Ist das nächste Programmelement ein Ausdruck, so wird dieser gemäß der unten stehenden Regeln in der aktuellen Umgebung zu einem Wert ausgewert. Ist das nächste Programmelement eine Konstantendefinition (**define** x e), so wird in der aktuellen Umgebung zunächst e zu einem Wert v ausgewertet und dann (**define** x v) zur aktuellen Umgebung hinzugefügt.

(FUN) $(name\ v_1 \dots v_n) \rightarrow e[name_1 := v_1 \dots name_n := v_n]$ falls (**define** ($name\ name_1 \dots name_n$) e) in Umgebung
 Skript 8.9.1

(PRIM) $(name\ v_1 \dots v_n) \rightarrow v$ falls $name$ eine primitive Funktion f ist und $f(v_1, \dots, v_n) = v$
 Skript 8.9.1

(CONST) $name \rightarrow v$ falls (**define** $name\ v$) in Umgebung
 Skript 8.9.2

(COND-True) $(\text{cond} [\#true\ e] \dots) \rightarrow e$
 Skript 8.9.3

(COND-False) $(\text{cond} [\#false\ e_1] [e_2\ e_3] \dots) \rightarrow (\text{cond} [e_2\ e_3] \dots)$
 Skript 8.9.3

(STRUCT-make) $(\text{make-name}\ v_1 \dots v_n) \rightarrow \langle \text{make-name}\ v_1 \dots v_n \rangle$ falls (**define-struct** $name\ (name_1 \dots name_n)$) in Umgebung
 Skript 8.9.5

(STRUCT-select) $(name\ name_i \langle \text{make-name}\ v_1 \dots v_n \rangle) \rightarrow v_i$ falls (**define-struct** $name\ (name_1 \dots name_n)$) in Umgebung
 Skript 8.9.5

(STRUCT-predtrue) $(name? \langle \text{make-name} \dots \rangle) \rightarrow \#true$
 Skript 8.9.5

(STRUCT-predfalse) $(name? v) \rightarrow false$ falls v nicht $\langle \text{make-name} \dots \rangle$ ist
 Skript 8.9.5

Äquivalenzregeln

(EKONG) $E[e_1] \equiv E[e_2]$ falls $e_1 \equiv e_2$
 Skript 8.12

(EREFL) $e \equiv e$
 Skript 8.12

(EKOMM) $e_2 \equiv e_1$ falls $e_1 \equiv e_2$
 Skript 8.12

(ETRANS) $e_1 \equiv e_3$ falls $e_1 \equiv e_2$ und $e_2 \equiv e_3$
 Skript 8.12

(ERED) $e_1 \equiv e_2$ falls $e_1 \rightarrow e_2$
 Skript 8.12

(EFUN) $(name\ e_1 \dots e_n) \equiv e[name_1 := e_1 \dots name_n := e_n]$ falls (**define** ($name\ name_1 \dots name_n$) e) in Umgebung
 Skript 8.12

$\langle E \rangle ::= []$
 $| (\langle name \rangle \langle e \rangle^* \langle E \rangle \langle e \rangle^*)$
 $| (\text{cond} \{ [\langle e \rangle \langle e \rangle] \}^* [\langle E \rangle \langle e \rangle] \{ [\langle e \rangle \langle e \rangle] \}^*)$
 $| (\text{cond} \{ [\langle e \rangle \langle e \rangle] \}^* [\langle e \rangle \langle E \rangle] \{ [\langle e \rangle \langle e \rangle] \}^*)$