

SECD(H)-Cheatsheet

SECD-AUSWERTUNGSREGELN

Erster Wert im *Code* entscheidet welche Regel angewandt wird!

1. Konstanten-Regel:

$$\langle S, E, bC, D \rangle \rightarrow_{SECD} \langle bS, E, C, D \rangle$$

2. Variablen-Regel:

$$\langle S, E, xC, D \rangle \rightarrow_{SECD} \langle E(x)S, E, C, D \rangle$$

3. Primitivum-Regel:

$$\langle T_n \dots T_1 S, E, \text{prim}_O C, D \rangle \rightarrow_{SECD} \langle O(T_1 \dots T_n) S, E, C, D \rangle$$

4. Abstraktions-Regel:

$$\langle S, E, \langle x, Y \rangle C, D \rangle \rightarrow_{SECD} \langle \langle \langle x, Y \rangle, E \rangle S, E, C, D \rangle$$

5. Applikations-Regel:

$$\langle T \langle \langle x, Y \rangle, E' \rangle S, E, apC, D \rangle \rightarrow \langle \epsilon, E' \cup \{x \rightarrow T\}, Y, \langle S, E, C, D \rangle \rangle$$

6. Abstraktion-Terminierungs-Regel:

$$\langle aS, E, \epsilon, \langle S', E', C', D' \rangle \rangle \rightarrow_{SECD} \langle aS', E', C', D' \rangle$$

7. Programm-Terminierungs-Regel:

$$\langle S, E, \epsilon, \epsilon \rangle \rightarrow_{SECD} \langle S, E, \epsilon, \epsilon \rangle$$

Programm-Ausführung ist nach diesem Schritt beendet!

Erster Wert auf dem *Stack* entspricht dem Rückgabewert!

λ -KALKÜL \Rightarrow ZWISCHENREPRÄSENTATION

1. Konstanten und Variablen:

$$\llbracket c \rrbracket_{SECD} = c$$

2. Abstraktionen:

$$\llbracket (\lambda x. Y) \rrbracket_{SECD} = \langle x, \llbracket Y \rrbracket_{SECD} \rangle$$

3. Applikationen:

$$\llbracket (Y x) \rrbracket_{SECD} = \llbracket Y \rrbracket \llbracket x \rrbracket ap$$

4. Primitive Operationen:

$$\llbracket (O^n b_1 \dots b_n) \rrbracket_{SECD} = \llbracket b_1 \rrbracket_{SECD} \dots \llbracket b_n \rrbracket_{SECD} \text{prim}_O$$

5. Zuweisung:

$$\llbracket (set! v e') \rrbracket_{SECDH} = v \llbracket e' \rrbracket_{SECDH} :=$$

SECDH-AUSWERTUNGSREGELN

Erster Wert im *Code* entscheidet welche Regel angewandt wird!

1. Konstanten-Regel:

$$\langle S, E, bC, D, H \rangle \rightarrow_{SECDH} \langle aS, E, C, D, H[a \mapsto b] \rangle$$

2. Variablen-Regel:

$$\langle S, E, xC, D, H \rangle \rightarrow_{SECDH} \langle E(x)S, E, C, D, H \rangle$$

3. Primitivum-Regel*:

$$\langle T_n \dots T_1 S, E, \text{prim}_O C, D, H \rangle \rightarrow \langle aS, E, C, D, H[a \mapsto O(T_1 \dots T_n)] \rangle$$

4. Abstraktions-Regel*:

$$\langle S, E, \langle x, Y \rangle C, D, H \rangle \rightarrow_{SECDH} \langle aS, E, Y, D, H[a \mapsto \langle \langle x, Y \rangle, E \rangle] \rangle$$

5. Applikations-Regel*:

$$\langle a_1 a_0 S, E, apC, D, H \rangle \rightarrow \langle \epsilon, E \cup \{v \mapsto a\}, H(a_0), \langle S, E, C, D \rangle, H[a \mapsto H(a_1)] \rangle$$

6. Abstraktions-Terminierungs-Regel:

$$\langle a, E, \epsilon, \langle S', E', C', D' \rangle, H \rangle \rightarrow_{SECDH} \langle S', E', C', D', H \rangle$$

7. Programm-Terminierungs-Regel:

$$\langle S, E, \epsilon, \epsilon, H \rangle \rightarrow_{SECD} \langle S, E, \epsilon, \epsilon, H \rangle$$

8. Zuweisungs-Regel:

$$\langle a_1 a_0 S, E, := C, D, H \rangle \rightarrow_{SECDH} \langle aS, E, C, D, H[a_0 \mapsto H(a_1)] [a \mapsto \text{void}] \rangle$$

*) a bezeichnet immer eine frische Adresse auf dem Heap!

SECD: $\langle \text{Stack}, \text{Environment}, \text{Code}, \text{Dump} \rangle$

SECDH: $\langle \text{Stack}, \text{Environment}, \text{Code}, \text{Dump}, \text{Heap} \rangle$

