5 Intermediate Language: SECK Machine

$$
\begin{array}{rcll}
Cmd & ::= & \texttt{FRAME} \mid \texttt{DEFRAME} & \\
 & \mid & \texttt{FETCH(}n\texttt{)} & \\
 & \mid & \texttt{QUOTE(}p\texttt{)} & \\
 & \mid & \texttt{ADD} \mid \texttt{SUB} \mid \texttt{MUL} \mid \texttt{DIV} & \\
 & \mid & \texttt{READ} \mid \texttt{WRITE} & \\
 & \mid & \texttt{COND(}Cmd^+\texttt{,}Cmd^+\texttt{)} & \\
 & \mid & \texttt{FTN(}Cmd^+\texttt{)} & \\
 & \mid & \texttt{APP} & \\
 & \mid & \texttt{RETURN} & \\
 & \mid & \texttt{UNROLL} & \\
p & ::= & n & \text{integer} \\
 & \mid & \mathit{true} \mid \mathit{false} & \text{boolean} \\
 & \mid & s & \text{string}
\end{array}
$$

SECK machine's state is a tuple

$$(S, E, C, K)$$

where

$$
\begin{array}{rcll}
\text{stack} & S & \in & \mathit{Stk} = \mathit{Value}\ \text{list} \\
\text{environment} & E & \in & \mathit{Env} = \mathit{Value}\ \text{list} \\
\text{code} & C & \in & \mathit{Code} = \mathit{Cmd}\ \text{list} \\
\text{continuation} & K & \in & \mathit{Cont} = (\mathit{Env} \times \mathit{Code})\ \text{list} \\
\text{value} & v & \in & \mathit{Value} = \mathit{Int} \cup \mathit{Bool} \cup \mathit{String} \cup \mathit{Code} \times \mathit{Env}
\end{array}
$$

SECK machine's execution is defined as the sequence of machine states linked by the following $\rightarrow$ transition:

$$
\begin{array}{rcl}
(v.S,\ E,\ \texttt{FRAME}.C,\ K) & \rightarrow & (S,\ v.E,\ C,\ K) \\
(S,\ v.E,\ \texttt{DEFRAME}.C,\ K) & \rightarrow & (S,\ E,\ C,\ K) \\
(S,\ E \text{ as } v_0.\cdots.v_n.E',\ \texttt{FETCH(}n\texttt{)}.C,\ K) & \rightarrow & (v_n.S,\ E,\ C,\ K) \\
(S,\ E,\ \texttt{QUOTE(}v\texttt{)}.C,\ K) & \rightarrow & (v.S,\ E,\ C,\ K) \\
(v_2.v_1.S,\ E,\ \texttt{ADD}.C,\ K) & \rightarrow & (v_1+v_2.S,\ E,\ C,\ K) \\
(S,\ E,\ \texttt{READ}.C,\ K) & \rightarrow & (\mathit{get}().S,\ E,\ C,\ K) \\
(v.S,\ E,\ \texttt{WRITE}.C,\ K) & \rightarrow & (v.S,\ E,\ C,\ K) \\
(\mathit{true}.S,\ E,\ \texttt{COND(}C_1\texttt{,}C_2\texttt{)}.C,\ K) & \rightarrow & (S,\ E,\ C_1.C,\ K) \\
(\mathit{false}.S,\ E,\ \texttt{COND(}C_1\texttt{,}C_2\texttt{)}.C,\ K) & \rightarrow & (S,\ E,\ C_2.C,\ K) \\
(S,\ E,\ \texttt{FTN(}C'\texttt{)}.C,\ K) & \rightarrow & ((C',\ E).S,\ E,\ C,\ K) \\
(v.(C',\ E').S,\ E,\ \texttt{APP}.C,\ K) & \rightarrow & (S,\ v.E',\ C',\ (E,\ C).K) \\
(S,\ E,\ \texttt{RETURN},\ (E',\ C).K) & \rightarrow & (S,\ E',\ C,\ K) \\
(S,\ v.E',\ \texttt{UNROLL}.C,\ K) & \rightarrow & (S,\ v.(\texttt{UNROLL}.C,\ E').E',\ C,\ K)
\end{array}
$$

5.1 Translation from tcoML to SECK

Each expression e of lexical depth ℓ is translated into SECK code C. We write this translation relation as

$$L, \ell, e \ \triangleright \ C.$$

The relation says that

- tcoML expression e is at lexical depth ℓ,
- e's free variable x's lexical depth is $L(x)$, and
- tcoML expression e is translated into SECK machine code C.

The generated SECK code C for tcoML expression e satisfies the invariant that the SECK machine, after executing the C and right before continuation, has the expression e's value on top of its stack S. The L is a table from identifiers to their lexical depths.

Rules for determining this translation relation are in Figure 1. For tcoML program e, its $SECK$ code is C such that

$$\emptyset, 0, e \ \triangleright \ C.$$

5.1.1 Correctness of the Translation

Theorem 6 *For tcoML program e, let $SECK$ code C be such that $\emptyset, 0, e \ \triangleright \ C$. Then $\emptyset \ \vdash \ e \ \Rightarrow \ v$ iff $([], [], C, []) \overset{*}{\rightarrow} (v, [], [], [])$.*

Let $L, \ell, e \ \triangleright \ C_e$ occurs during $\emptyset, 0, p \ \triangleright \ C$. Then followings are true:

Fact 2 *Expression e's lexical depth ℓ is the number of frames in E right before executing the code C_e.*

Fact 3 *The ℓ frames are those that were pushed in by function calls or entering let-blocks.*

Fact 4 *Calling a function (`fn x =>` $\cdots$) or entering a let-block (`let` [`rec`] `x =` $\cdots$) pushes one frame for the bound variable x.*

Fact 5 *Variable x's lexical depth $L(x)$ is the index of its frame in the current E.*

$$[\mathrm{C}_{\triangleright}] \qquad L, \ell, \mathit{const} \ \triangleright\ \mathtt{QUOTE(}\mathit{const}\mathtt{)}$$

$$[\mathrm{X}_{\triangleright}] \qquad L, \ell, x \ \triangleright\ \mathtt{FETCH(}\ell - L(x)\mathtt{)}$$

$$[\mathrm{FN}_{\triangleright}] \qquad \frac{L[x \mapsto \ell+1], \ell+1, e \ \triangleright\ C}{L, \ell, \mathtt{fn}\ x\ \mathtt{=>}\ e \ \triangleright\ \mathtt{FTN(}C\mathtt{.RETURN)}}$$

$$[\mathrm{APP}_{\triangleright}] \qquad \frac{L, \ell, e_1 \ \triangleright\ C_1 \qquad L, \ell, e_2 \ \triangleright\ C_2}{L, \ell, e_1\ e_2 \ \triangleright\ C_1.C_2.\mathtt{APP}}$$

$$[\mathrm{LET}_{\triangleright}] \qquad \frac{L, \ell, e_1 \ \triangleright\ C_1 \qquad L[x \mapsto \ell+1], \ell+1, e_2 \ \triangleright\ C_2}{L, \ell, \mathtt{let}\ x\ \mathtt{=}\ e_1\ \mathtt{in}\ e_2\ \mathtt{end} \ \triangleright\ C_1.\mathtt{FRAME}.C_2.\mathtt{DEFRAME}}$$

$$[\mathrm{RLET}_{\triangleright}] \qquad \frac{L[f \mapsto \ell+1][x \mapsto \ell+2], \ell+2, e_1 \ \triangleright\ C_1 \qquad L[f \mapsto \ell+1], \ell+1, e_2 \ \triangleright\ C_2}{L, \ell, \mathtt{let\ rec}\ f\ \mathtt{=}\ \mathtt{fn}\ x\ \mathtt{=>}\ e_1\ \mathtt{in}\ e_2\ \mathtt{end} \ \triangleright\ \begin{array}{l}\mathtt{FTN(UNROLL.}C_1\mathtt{.RETURN).}\\ \mathtt{FRAME.}C_2\mathtt{.DEFRAME}\end{array}}$$

$$[\mathrm{IF}_{\triangleright}] \qquad \frac{L, \ell, e_i \ \triangleright\ C_i}{L, \ell, \mathtt{if}\ e_1\ \mathtt{then}\ e_2\ \mathtt{else}\ e_3 \ \triangleright\ C_1.\mathtt{COND(}C_2\mathtt{,}\ C_3\mathtt{)}}$$

$$[\mathrm{ADD}_{\triangleright}] \qquad \frac{L, \ell, e_i \ \triangleright\ C_i}{L, \ell, e_1 \,\mathtt{+}\, e_2 \ \triangleright\ C_1.C_2.\mathtt{ADD}}$$

$$[\mathrm{OR}_{\triangleright}] \qquad \frac{L, \ell, e_i \ \triangleright\ C_i}{L, \ell, e_1\ \mathtt{or}\ e_2 \ \triangleright\ C_1.\mathtt{COND(QUOTE(}\mathit{true}\mathtt{),}C_2\mathtt{)}}$$

$$[\mathrm{AND}_{\triangleright}] \qquad \frac{L, \ell, e_i \ \triangleright\ C_i}{L, \ell, e_1\ \mathtt{and}\ e_2 \ \triangleright\ C_1.\mathtt{COND(}C_2\mathtt{,QUOTE(}\mathit{false}\mathtt{))}}$$

$$[\mathrm{IN}_{\triangleright}] \qquad L, \ell, \mathtt{read} \ \triangleright\ \mathtt{READ}$$

$$[\mathrm{OUT}_{\triangleright}] \qquad \frac{L, \ell, e \ \triangleright\ C_1}{L, \ell, \mathtt{write}\ e \ \triangleright\ C_1.\mathtt{WRITE}}$$

그림 1: Translation from tcoML to SECK