The SECDH_1-machine instructions redefined

$$(S,\ pe,\ E,\ pc,\ C[\ pc \leadsto \texttt{PUSH\_S}\ atom\ ],\ D,\ H\ ) \\ \rightarrow\ \ (atom : S,\ pe,\ E,\ pc+1,\ C[],\ D,\ H\ )$$

$$(S,\ pe,\ E[\ deref(\ pe,\ i) \leadsto\ ppf : se_1 : \ldots : se_{(j+1)} : \ldots se_n\ ] \\ pc,\ C[\ pc \leadsto \texttt{PUSH\_ES}\ i\ j\ ],\ D,\ H\ ) \\ \rightarrow\ \ (se_{(j+1)} : S,\ pe,\ E[],\ pc+1,\ C[],\ D,\ H\ )$$

$$(ppf : se_1 : \ldots : se_n : S,\ pe,\ E,\ pc,\ C[\ pc \leadsto \texttt{MKFRAME}\ n\ ],\ D,\ H\ ) \\ \rightarrow\ (S,\ pe' = pe+n+1,\ pe' \leadsto\ ppf : se_n : \ldots : se_1 : E, \\ pc+1,\ C[],\ D,\ H\ )$$

$$(S,\ pe,\ E,\ pc,\ C[\ pc \leadsto \texttt{FREE}\ n\ ],\ D,\ H\ ) \\ \rightarrow\ \ (S,\ pe-n-1,\ E,\ pc+1,\ C[],\ D,\ H\ )$$

$$(\#t \mid \#f : S,\ pe,\ E,\ pc,\ C[\ pc \leadsto \texttt{BRC}\ p\_t \mid p\_f\ ],\ D,\ H\ ) \\ \rightarrow\ \ (S,\ pe,\ E,\ p\_t \mid p\_f,\ C[],\ (pc+1) : D,\ H\ )$$

$$(S,\ pe,\ E,\ pc,\ C[\ pc \leadsto \texttt{CALL}\ p\_ff\ ],\ D,\ H\ ) \\ \rightarrow\ \ (S,\ pe,\ E,\ p\_ff,\ C[],\ (pc+1) : D,\ H\ )$$

$$(p\_ff : S,\ pe,\ E,\ pc,\ C[\ pc \leadsto \texttt{AP}\ r \mid p\_ff \leadsto \texttt{ARGS}\ n\ ],\ D,\ H\ ) \\ \rightarrow\ \begin{cases} (S,\ pe,\ E,\ p\_ff,\ C[],\ (pc+1) : D,\ H\ ) & \text{if } n = r \\ (error\_message, -, -, -, -\ ) & \text{otherwise} \end{cases}$$

$$(S,\ pe,\ E,\ pc,\ C[\ pc \leadsto \texttt{RET}\ ],\ ppc : D,\ H\ ) \\ \rightarrow\ \ (S,\ pe,\ E,\ ppc,\ C[],\ D,\ H\ )$$

New instructions that need be added

$$(\ S, \ pe, \ E, \ pc, \ C[\,pc \leadsto \texttt{LINK}\, h\,], \ D, \ H \ ) \\ \rightarrow \ (\ p\_ch : S, \ pe, \ E, \ pc+1, \ C[], \ D, \ H \ )$$

$$\text{where} \quad p\_ch \ = \ \begin{cases} pe \quad \text{if} \quad h = -1 \\ deref(\,pe, \ i+1\,) \quad \text{if} \quad h \geq 0 \end{cases}$$

$$(\ S, \ pe, \ E, \ pc, \ C[\,pc \leadsto \texttt{ARGS}\, n\,], \ D, \ H \ ) \\ \rightarrow \ (\ S, \ pe, \ E, \ pc+1, \ C[] \ D, \ H \ )$$

$$(\ S, \ pe, \ E, \ pc, \ C[\,pc \leadsto \texttt{CREATE\_CLOS}\, h \ ff\,], \ D, \ H \ ) \\ \rightarrow \ (\ p\_clos : S, \ pe, \ E, \ pc+1, \ C[], \\ D, \ p\_clos \leadsto < \ pp, \ p\_ff \ >: H \ )$$

$$\text{where} \quad p\_ch \ = \ \begin{cases} pe \quad \text{if} \quad h = -1 \\ deref(\,pe, \ i+1\,) \quad \text{if} \quad h \geq 0 \end{cases}$$

$$(\ p\_clos : S, \ pe, \ E, \ pc, \\ C[\,pc \leadsto \texttt{AP}\ r \mid p\_ff \leadsto \texttt{ARGS}\, n\,], \\ D, \ H[\,p\_clos \leadsto < \ pp, \ p\_ff \ >\,] \ ) \\ \rightarrow \ \begin{cases} (\ pp : S, \ pe, \ E, \ p\_ff, \ C[], \ (pc+1) : D, \ H[] \ ) \quad \text{if} \ n = r \\ (\ error\_message, \ -, -, -, -, -, - \ ) \quad \text{otherwise} \end{cases}$$

Note that

$\rightarrow$ $deref(\ pe,\ i+1\ )$ dereferences i times the chain of frame pointers and returns the frame pointer found in the frame thus referenced.

$\rightarrow$ `ARGS` n is a pseudo-instruction without an effect $\leadsto$ it merely serves to check, in conjunction with the instruction `AP` r, the arity of function code.

Modified compilation rules

$$\mathcal{C}[\ [\ \#i\ \#j\ ] : es\ ] \ \rightarrow\ \ \texttt{PUSH\_ES}\ i\ j;\ \mathcal{C}[\ es\ ];$$

$$\begin{aligned}\mathcal{C}[\ \Lambda^{(r)} e_b : ap^{(r)} : es\ ] \ \rightarrow&\ \ \texttt{LINK}\ -1;\ \texttt{CALL}\ p\_ff;\ \mathcal{C}[\ es\ ];\\ p\_ff \leadsto&\ \ \mathcal{F}[\ \Lambda^{(r)} e_b\ ];\end{aligned}$$

$$\begin{aligned}\mathcal{C}[\ \Lambda^{(r)} e_b : es\ ] \ \rightarrow&\ \ \texttt{CREATE\_CLOS}\ -1\ p\_ff;\ \mathcal{C}[\ es\ ];\\ p\_ff \leadsto&\ \ \mathcal{F}[\ \Lambda^{(r)} e_b\ ];\end{aligned}$$

$$\begin{aligned}&\mathcal{C}[\ [\ \#h\ ff\ ] : ap^{(r)} : es\ ] \mid p\_ff \leadsto\ \Lambda^{(r)} e_b\\ &\qquad\qquad \rightarrow\ \ \texttt{LINK}\ h;\ \texttt{CALL}\ p\_ff;\ \mathcal{C}[\ es\ ];\end{aligned}$$

$$\mathcal{C}[\ [\ \#h\ ff\ ] : es\ ] \ \rightarrow\ \ \texttt{CREATE\_CLOS}\ h\ p\_ff;\ \mathcal{C}[\ es\ ];$$

$$\mathcal{F}[\ e\ ] \ \rightarrow\ \begin{cases} \texttt{ARGS}\ r;\ \texttt{MKFRAME}\ r;\ \mathcal{C}[\ e_b\ ];\ \texttt{FREE}\ r;\ \texttt{RET};\\ \text{if}\ \ e = \Lambda^{(r)} e_b \\ \\ \mathcal{C}[\ e\ ] \quad \text{otherwise} \end{cases}$$

All other compilation rules remain the same as for the SECDH_1-machine