

# The SECD Virtual Machine

The SECD Virtual Machine is the great grandfather of the Java Virtual Machine (JVM).

- S Stack: the operand stack, contains values.
- E Environment: maps variables to values.
- C Control: the sequence of instructions to be performed.
- D Dump: the procedure call stack.

The machine has three kinds of instructions for 1) referencing a variable, 2) creating a function, and 3) applying (calling) a function.

$$I ::= \text{var } x \mid (\lambda x. C) \mid \text{app}$$

# The SECD Virtual Machine

The following function *compile* translates an lambda calculus expression into a sequence of instructions for the SECD machine.

$$\begin{aligned} \text{compile}(x) &= [\text{var } x] \\ \text{compile}((e_1 e_2)) &= \text{compile}(e_1) \cdot \text{compile}(e_2) \cdot \text{app} \\ \text{compile}((\lambda x. e)) &= [(\lambda x. \text{compile}(e))] \end{aligned}$$

(The notation  $[a]$  is a single-element sequence whose element is  $a$ . The notation  $a \cdot ls$  adds the element  $a$  to the front of the sequence  $ls$ .)

# The SECD Virtual Machine

- ▶ variable lookup

$$(S, E, x \cdot C, D) \mapsto (E(x) \cdot S, E, C, D)$$

- ▶ closure creation

$$(S, E, (\lambda x. C') \cdot C, D) \mapsto (\langle \lambda x. C', E \rangle \cdot S, E, C, D)$$

- ▶ function application

$$(v \cdot \langle \lambda x. C', E' \rangle \cdot S, E, app \cdot C, D) \mapsto ([], E'(x \rightarrow v), C', (S, E, C, D))$$

- ▶ function return

$$(v \cdot S, E, [], (S', E', C', D)) \mapsto (v \cdot S', E', C', D)$$

# The CEK Abstract Machine

$c ::= \langle e, E \rangle$  closures  
 $k ::= [] \mid \text{arg } c \ k \mid \text{fun } v \ k$  continuations

- ▶ variable lookup

$$(\langle x, E \rangle, k) \mapsto (E(x), k)$$

- ▶ application: start evaluating the function expression

$$(\langle (e_1 \ e_2), E \rangle, k) \mapsto (\langle e_1, E \rangle, \text{arg } \langle e_2, E \rangle \ k)$$

- ▶ application: start evaluating the argument expression

$$(\langle v, E \rangle, \text{arg } \langle e, E' \rangle \ k) \mapsto (\langle e, E' \rangle, \text{fun } v \ E \ k)$$

- ▶ function call: start evaluating the body

$$(\langle v, E \rangle, \text{fun } \langle (\lambda x. e), E' \rangle \ k) \mapsto (\langle e, E'(x \rightarrow \langle v, E \rangle) \rangle, k)$$