CC, SCC, CK, and CEK Machines

Summary and Refinements

CS 6520, Spring 2000

CC Machine

Specialized for the case that all primitives are binary operators:

$\langle (M\ N), E[\,]\rangle$ if $M \notin Vals$	$\longmapsto$	$\langle M, E[([\,]\ N)]\rangle$	1
$\langle (V\ M), E[\,]\rangle$ if $M \notin Vals$	$\longmapsto$	$\langle M, E[(V\ [\,])]\rangle$	2
$\langle (o\ M\ N), E[\,]\rangle$ if $M \notin Vals$	$\longmapsto$	$\langle M, E[(o\ [\,]\ N)]\rangle$	$3a$
$\langle (o\ V\ M), E[\,]\rangle$ if $M \notin Vals$	$\longmapsto$	$\langle M, E[(o\ V\ [\,])]\rangle$	$3b$
$\langle ((\lambda x.M)\ V), E[\,]\rangle$	$\longmapsto$	$\langle M[V/x], E[\,]\rangle$	β_v
$\langle (o\ b_1\ b_2), E[\,]\rangle$	$\longmapsto$	$\langle V, E[\,]\rangle$ where $V = \delta(o, b_1, b_2)$	δ_v
$\langle V, E[(U\ [\,])]\rangle$	$\longmapsto$	$\langle (U\ V), E[\,]\rangle$	4
$\langle V, E[([\,]\ N)]\rangle$	$\longmapsto$	$\langle (V\ N), E[\,]\rangle$	5
$\langle V, E[(o\ U\ [\,])]\rangle$	$\longmapsto$	$\langle (o\ U\ V), E[\,]\rangle$	$6a$
$\langle V, E[(o\ [\,]\ N)]\rangle$	$\longmapsto$	$\langle (o\ V\ N), E[\,]\rangle$	$6b$

SCC Machine

Still specialized for binary-only operators:

$\langle (M\ N), E[\,]\rangle$	$\longmapsto$	$\langle M, E[([\,]\ N)]\rangle$	1
$\langle (o\ M\ N), E[\,]\rangle$	$\longmapsto$	$\langle M, E[(o\ [\,]\ N)]\rangle$	2
$\langle V, E[((\lambda x.M)\ [\,])]\rangle$	$\longmapsto$	$\langle M[V/x], E[\,]\rangle$	3
$\langle V, E[([\,]\ N)]\rangle$	$\longmapsto$	$\langle N, E[(V\ [\,])]\rangle$	4
$\langle b_2, E[(o\ b_1\ [\,])]\rangle$	$\longmapsto$	$\langle V, E[\,]\rangle$ where $V = \delta(o, b_1, b_2)$	5
$\langle V, E[(o\ [\,]\ N)]\rangle$	$\longmapsto$	$\langle N, E[(o\ V\ [\,])]\rangle$	6

CK Machine

Since we only have binary primitive operators, define K as:

$$
\begin{array}{rcll}
K & ::= & \mathsf{mt} & \text{nothing left to do} \\
 & | & \langle \mathsf{fun}, V, K \rangle & \text{apply function} \\
 & | & \langle \mathsf{arg}, N, K \rangle & \text{evaluate argument (then apply function)} \\
 & | & \langle \mathsf{prim}, o, V, K \rangle & \text{apply primitive} \\
 & | & \langle \mathsf{primarg}, o, N, K \rangle & \text{evaluate other primitive argument (then apply primitive)}
\end{array}
$$

Our CK reduction rules are:

$$
\begin{array}{lcll}
\langle (M\ N), K \rangle & \longmapsto & \langle M, \langle \mathsf{arg}, N, K \rangle \rangle & 1 \\
\langle (o\ M\ N), K \rangle & \longmapsto & \langle M, \langle \mathsf{primarg}, o, N, K \rangle \rangle & 2 \\
\langle V, \langle \mathsf{fun}, (\lambda x.M), K \rangle \rangle & \longmapsto & \langle M[V/x], K \rangle & 3 \\
\langle V, \langle \mathsf{arg}, N, K \rangle \rangle & \longmapsto & \langle N, \langle \mathsf{fun}, V, K \rangle \rangle & 4 \\
\langle b_2, \langle \mathsf{prim}, o, b_1, K \rangle \rangle & \longmapsto & \langle V, K \rangle & 5 \\
 & & \quad \text{where } V = \delta(o, b_1, b_2) & \\
\langle V, \langle \mathsf{primarg}, o, N, K \rangle \rangle & \longmapsto & \langle N, \langle \mathsf{prim}, o, V, K \rangle \rangle & 6
\end{array}
$$

Implementation

We can implement K with a set of structure declarations:

```
(define-struct mtK ())
(define-struct funK (vcl k))
(define-struct argK (ncl k))
(define-struct primargK (o ncl k))
(define-struct primK (o vcl k))
```

For easy printing of these structures, we turn on structure printing in DrScheme/MzScheme:

```
(print-struct #t)
```

CEK Machine

Our CEK reduction rules are:

$$
\begin{array}{lcll}
\langle \langle (M\ N), E \rangle, K \rangle & \longmapsto & \langle \langle M, E \rangle, \langle \mathsf{arg}, \langle N, E \rangle, K \rangle \rangle & 1 \\
\langle \langle (o\ M\ N), E \rangle, K \rangle & \longmapsto & \langle \langle M, E \rangle, \langle \mathsf{primarg}, o, \langle N, E \rangle, K \rangle \rangle & 2 \\
\langle \langle V, E \rangle, \langle \mathsf{fun}, \langle (\lambda x.M), E' \rangle, K \rangle \rangle & \longmapsto & \langle \langle M, E'[x \leftarrow \langle V, E \rangle] \rangle, K \rangle & 3 \\
\quad \text{if } V \notin Vars & & & \\
\langle \langle V, E \rangle, \langle \mathsf{arg}, \langle N, E' \rangle, K \rangle \rangle & \longmapsto & \langle \langle N, E' \rangle, \langle \mathsf{fun}, \langle V, E \rangle, K \rangle \rangle & 4 \\
\quad \text{if } V \notin Vars & & & \\
\langle \langle b_2, E_2 \rangle, \langle \mathsf{prim}, o, \langle b_1, E_1 \rangle, K \rangle \rangle & \longmapsto & \langle \langle V, \{\} \rangle, K \rangle & 5 \\
 & & \quad \text{where } V = \delta(o, b_1, b_2) & \\
\langle \langle V, E \rangle, \langle \mathsf{primarg}, o, \langle N, E' \rangle, K \rangle \rangle & \longmapsto & \langle \langle N, E' \rangle, \langle \mathsf{prim}, o, \langle V, E \rangle, K \rangle \rangle & 6 \\
\quad \text{if } V \notin Vars & & & \\
\langle \langle x, E \rangle, K \rangle & \longmapsto & \langle \langle V, E' \rangle, K \rangle & 7 \\
 & & \quad \text{where } \langle V, E' \rangle = E(x) &
\end{array}
$$