The CSE Machine

Programming Languages
Lecture 7

Adeesha Wijayasiri

Evaluation of RPAL Programs

- Need an algorithm to complete the operational semantic specification of RPAL

Introducing the CSE Machine

- C - Control
  - a sequence of operations
- S - Stack
  - operands
- E - Environment
  - Initially, PE (Primitive Environment)
  - Updated as evaluation proceeds

- PE: a mapping from names to objects and operations.

CSE Machine programs: control structures

- Flatten the RPAL program's ST into a "control structure“
- Done using a simple pre-order tree traversal.

Example

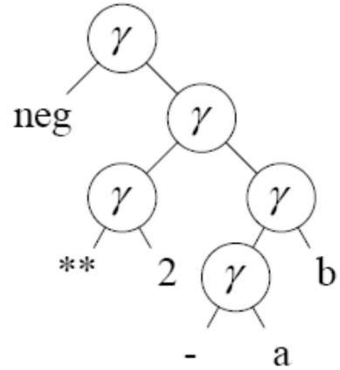
- Evaluate -2 ** (a-b), in an environment in which (somehow) a=6 and b=1.

- Flattened control structure:

  γ neg γ γ ** 2 γ γ - a b.

- Place this control structure on the Control of the CSE Machine.

Example: Evaluate -2 ** (a-b), in an environment in which a=6 and b=1.



CONTROL	STACK	ENV
γ neg γ γ ** 2 γ γ - a b		PE
γ neg γ γ ** 2 γ γ - a	1	
γ neg γ γ ** 2 γ γ -	6 1	
γ neg γ γ ** 2 γ γ	Minus 6 1	
γ neg γ γ ** 2 γ	Minus6 1	
γ neg γ γ ** 2	5	
γ neg γ γ **	2 5	
γ neg γ γ	Exp 2 5	
γ neg γ	Exp2 5	
γ neg	32	
γ	Neg 32	
	-32	

CSE Machine Operation (informally)

1. Remove right-most item from control.
2. If a name, look it up in the CE (current environment), push onto the stack.
3. If γ, then
   - rator = pop(stack)
   - rand = pop(stack)
   - push(apply(rator,rand), stack)
4. Stop if control is empty: value on the stack is the result.

Notes

- Minus: function that subtracts its second argument from its first one.
- Minus6: a function that subtracts its argument from 6.
- Exp, likewise: the exponentiation function.
- Exp2: function that raises 2 to the power of its argument.

Notes (cont'd)

- Notice difference between "neg" (a name), and "Neg" (the actual operator).
- Control contains gammas (and lambdas) and names. Stack contains "real" values.

Generating Control Structures

- Begin with CS (control structure) δ_0:
- Perform a pre-order traversal of the standardized tree.
- For each node:
  a. If a name, add it to the current CS.
  b. If a γ, add it to the current CS.
  c. If a λ, add $<\lambda\ k\ x>$ to the current CS.
     - k: new index; x: λ's left child.
     - Generate control structure δ_k: traverse the λ's right child.

Generating Control Structures

- We use a single symbol to represent a λ-expression, both on the control, and on the stack. The symbol is $<i \lambda k\ x>$.
  - i: environment,
  - k: CS of the function's body,
  - x: the function's bound variable.

- The λ-expression becomes a λ-closure when its environment is determined, when it is placed on the stack.

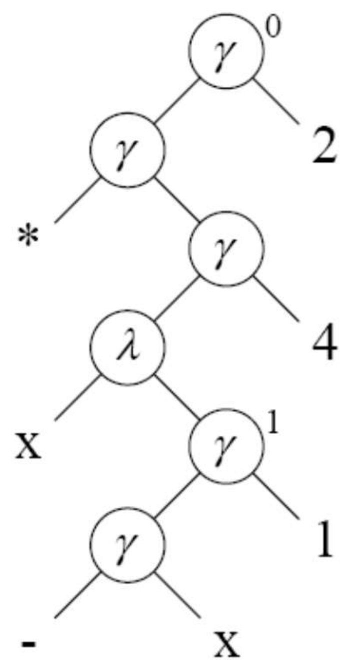
Examples

- Three examples of generating control structures.

Examples of Control Structure Generation:

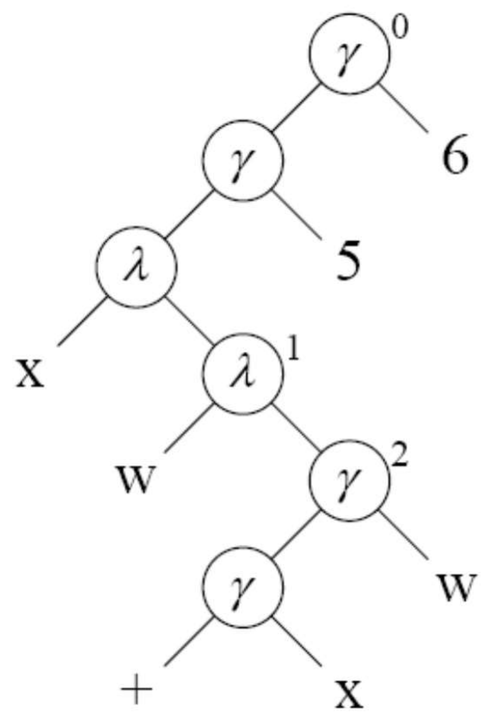
Example 1:

Applicative expression	Control Structures
$(\lambda x.x\text{-}1)4 * 2$	$\delta_0 = \gamma\ \gamma\ *\ \gamma\ \lambda_1^x\ 4\ 2$ $\delta_1 = \gamma\ \gamma\ -\ x\ 1$



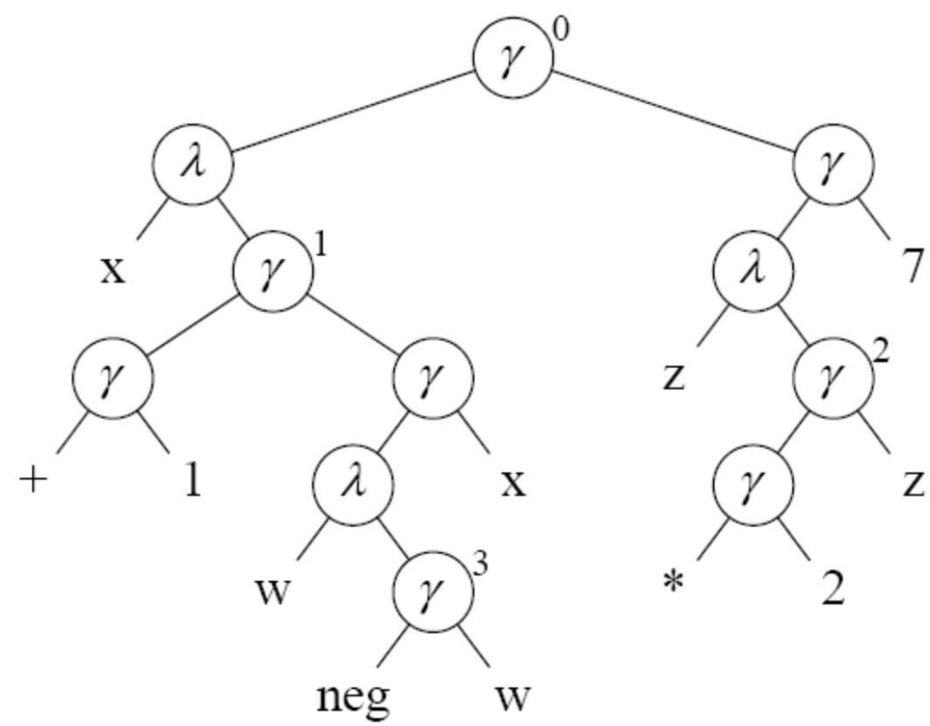
Example 2:

Applicative expression	Control Structures
$(\lambda x.\lambda w.x+w)\ 5\ 6$	$\delta_0 = \gamma\ \gamma\ \lambda_1^x\ 5\ 6$ $\delta_1 = \lambda_2^w$ $\delta_2 = \gamma\ \gamma\ +\ x\ w$



Example 3:

Applicative expression	Control Structures
$(\lambda x.1+(\lambda w.-w)x)[(\lambda z.2*z)7]$	$\delta_0 = \gamma\ \lambda_1^x\ \gamma\ \lambda_2^z\ 7$
	$\delta_1 = \gamma\ \gamma + 1\ \gamma\ \lambda_3^w\ x$
	$\delta_2 = \gamma\ \gamma * 2\ z$
	$\delta_3 = \gamma\ \text{neg}\ w$



Operation of the CSE Machine

- Five rules
- Process driven by TOP symbol on the control.
- Need environment markers, on the Control and Stack.
- Every environment is linked to a previously created (but not necessarily currently active) environment.
- Thus, environment structure is a tree.

CSE Machine Rules:

	CONTROL	STACK	ENV
Initial State	$e_0\ \delta_0$	e_0	e_0 = PE
CSE Rule 1 (stack a name)	.... Name	.... Ob	Ob=Lookup(Name,e_c) e_c:current environment
CSE Rule 2 (stack λ)	.... λ_k^x	.... ${}^c\lambda_k^x$	e_c:current environment
CSE Rule 3 (apply rator)	.... γ	Rator Rand .... Result	Result=Apply[Rator,Rand]
CSE Rule 4 (apply λ)	.... γ .... $e_n\ \delta_k$	${}^c\lambda_k^x$ Rand .... e_n	e_n = [Rand/x]e_c
CSE Rule 5 (exit env.)	.... e_n	value e_n .... value	

Examples of CSE Machine Operation

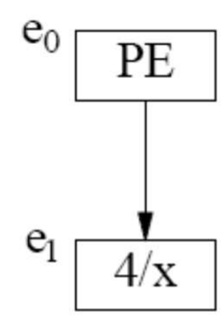
- Let’s run through the CSE machine, for our 3 examples.

Example 1:

Applicative expression	Control Structures
$(\lambda x.x\text{-}1)4 * 2$	$\delta_0 = \gamma\ \gamma\ *\ \gamma\ \lambda_1^x\ 4\ 2$ $\delta_1 = \gamma\ \gamma\ -\ x\ 1$

RULE	CONTROL	STACK	ENV
1	$e_0\ \gamma\ \gamma\ *\ \gamma\ \lambda_1^x\ 4\ 2$	e_0	e_0 = PE
1	$e_0\ \gamma\ \gamma\ *\ \gamma\ \lambda_1^x\ 4$	$2\ e_0$	
2	$e_0\ \gamma\ \gamma\ *\ \gamma\ \lambda_1^x$	$4\ 2\ e_0$	
4	$e_0\ \gamma\ \gamma\ *\ \gamma$	${}^0\lambda_1^x\ 4\ 2\ e_0$	
1	$e_0\ \gamma\ \gamma\ *\ e_1\ \gamma\ \gamma\ -\ x\ 1$	$e_1\ 2\ e_0$	$e_1 = [4/x]e_0$
1	$e_0\ \gamma\ \gamma\ *\ e_1\ \gamma\ \gamma\ -\ x$	$1\ e_1\ 2\ e_0$	
1	$e_0\ \gamma\ \gamma\ *\ e_1\ \gamma\ \gamma\ -$	$4\ 1\ e_1\ 2\ e_0$	
3	$e_0\ \gamma\ \gamma\ *\ e_1\ \gamma\ \gamma$	$-\ 4\ 1\ e_1\ 2\ e_0$	
3	$e_0\ \gamma\ \gamma\ *\ e_1\ \gamma$	$(-4)\ 1\ e_1\ 2\ e_0$	
5	$e_0\ \gamma\ \gamma\ *\ e_1$	$3\ e_1\ 2\ e_0$	
1	$e_0\ \gamma\ \gamma\ *$	$3\ 2\ e_0$	
3	$e_0\ \gamma\ \gamma$	$*\ 3\ 2\ e_0$	
3	$e_0\ \gamma$	$(*3)\ 2\ e_0$	
5	e_0	$6\ e_0$	
		6	

Environment tree:

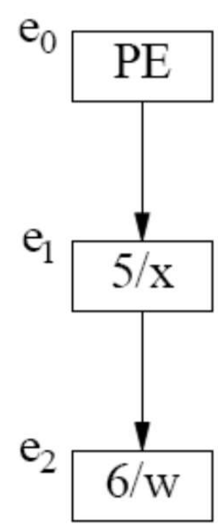


Example 2:

Applicative expression	Control Structures
$(\lambda x.\lambda w.x+w)\ 5\ 6$	$\delta_0 = \gamma\ \gamma\ \lambda_1^x\ 5\ 6$
	$\delta_1 = \lambda_2^w$
	$\delta_2 = \gamma\ \gamma\ +\ x\ w$

RULE	CONTROL	STACK	ENV
1	$e_0 \gamma\ \gamma\ \lambda_1^x\ 5\ 6$	e_0	e_0=PE
1	$e_0 \gamma\ \gamma\ \lambda_1^x\ 5$	$6\ e_0$	
2	$e_0 \gamma\ \gamma\ \lambda_1^x$	$5\ 6\ e_0$	
4	$e_0 \gamma\ \gamma$	${}^0\lambda_1^x\ 5\ 6\ e_0$	
2	$e_0 \gamma\ e_1\ \lambda_2^w$	$e_1\ 6\ e_0$	$e_1=[5/x]e_0$
5	$e_0 \gamma\ e_1$	${}^1\lambda_2^w\ e_1\ 6\ e_0$	
4	$e_0 \gamma$	${}^1\lambda_2^w\ 6\ e_0$	
1	$e_0\ e_2\ \gamma\ \gamma\ +\ x\ w$	$e_2\ e_0$	$e_2=[6/w]e_1$
1	$e_0\ e_2\ \gamma\ \gamma\ +\ x$	$6\ e_2\ e_0$	
1	$e_0\ e_2\ \gamma\ \gamma\ +$	$5\ 6\ e_2\ e_0$	
3	$e_0\ e_2\ \gamma\ \gamma$	$+\ 5\ 6\ e_2\ e_0$	
3	$e_0\ e_2\ \gamma$	$(+5)\ 6\ e_2\ e_0$	
5	$e_0\ e_2$	$11\ e_2\ e_0$	
5	e_0	$11\ e_0$	
		11	

Environment Structure:

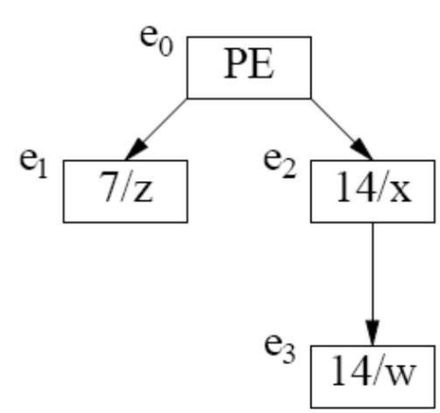


Example 3:

Applicative expression	Control Structures
$(\lambda x.1+(\lambda w.-w)x)[(\lambda z.2*z)7]$	$\delta_0 = \gamma\ \lambda_1^x\ \gamma\ \lambda_2^z\ 7$ $\delta_1 = \gamma\ \gamma\ +\ 1\ \gamma\ \lambda_3^w\ x$ $\delta_2 = \gamma\ \gamma\ *\ 2\ z$ $\delta_3 = \gamma\ \text{neg}\ w$

RULE	CONTROL	STACK	ENV
1	$e_0\ \gamma\ \lambda_1^x\ \gamma\ \lambda_2^z\ 7$	e_0	e_0=PE
2	$e_0\ \gamma\ \lambda_1^x\ \gamma\ \lambda_2^z$	$7\ e_0$	
4	$e_0\ \gamma\ \lambda_1^x\ \gamma$	${}^0\lambda_2^z\ 7\ e_0$	
1,1,1	$e_0\ \gamma\ \lambda_1^x\ e_1\ \gamma\ \gamma\ *\ 2\ z$	$e_1\ e_0$	$e_1=[7/z]e_0$
3,3	$e_0\ \gamma\ \lambda_1^x\ e_1\ \gamma\ \gamma$	$*\ 2\ 7\ e_1\ e_0$	
5	$e_0\ \gamma\ \lambda_1^x\ e_1$	$14\ e_1\ e_0$	
2	$e_0\ \gamma\ \lambda_1^x$	$14\ e_0$	
4	$e_0\ \gamma$	${}^0\lambda_1^x\ 14\ e_0$	
1	$e_0\ e_2\ \gamma\ \gamma\ +\ 1\ \gamma\ \lambda_3^w\ x$	$e_2\ e_0$	$e_2=[14/x]e_0$
2	$e_0\ e_2\ \gamma\ \gamma\ +\ 1\ \gamma\ \lambda_3^w$	$14\ e_2\ e_0$	
4	$e_0\ e_2\ \gamma\ \gamma\ +\ 1\ \gamma$	${}^2\lambda_3^w\ 14\ e_2\ e_0$	
1,1,3	$e_0\ e_2\ \gamma\ \gamma\ +\ 1\ e_3\ \gamma\ \text{neg}\ w$	$e_3\ e_2\ e_0$	$e_3=[14/w]e_2$
5	$e_0\ e_2\ \gamma\ \gamma\ +\ 1\ e_3$	$-14\ e_3\ e_2\ e_0$	
1,1,3,3	$e_0\ e_2\ \gamma\ \gamma\ +\ 1$	$-14\ e_2\ e_0$	
5	$e_0\ e_2$	$-13\ e_2\ e_0$	
5	e_0	$-13\ e_0$	
		-13	

Environment Structure:



Five CSE Rules (Minimally) Sufficient

- Let's take some shortcuts.
- CSE Rules 6 and 7: Unary and Binary Operators.

Five CSE Rules (Minimally) Sufficient (cont'd)

- In the control structures, abbreviate:

$\gamma\,\gamma$ +	to	+
$\gamma\,\gamma$ -	to	-
. . .	(other binary operators)	
γ neg	to	neg
γ not	to	not

- In other words, DO NOT standardize unops and binops.

Optimizations for the CSE Machine.

CSE Rules 6 and 7: Unary and Binary Operators.

	CONTROL	STACK	ENV
CSE Rule 6 (binop)	.... binop	Rand Rand	
	....	Result	Result=Apply[binop,Rand,Rand]
CSE Rule 7 (unop)	.... unop	Rand	
	....	Result	Result=Apply[unop,Rand]

CSE Rule 8: Conditional

- Do not standardize $\rightarrow$ node. Instead, for B $\rightarrow$ E1 | E 2, generate

  $\delta_{then}\ \delta_{else}\ \beta$ B, where

  δ_{then} = control structure for E1

  δ_{else} = control structure for E2

- B evaluated first, then β pops the stack, keeps one δ and discards the other.

CSE Rule 8: Conditional.

	CONTROL	STACK	ENV
CSE Rule 8 (Conditional)	.... δ_{then} δ_{else} β	true	
	.... δ_{then}	....	
	.... δ_{then} δ_{else} β	false	
	.... δ_{else}	....	

Example:

Applicative expression	Control Structures
$(\lambda n.n<0 \rightarrow -n \mid n)\ (-3)$	$\delta_0 = \gamma\ \lambda_1^n$ neg 3
	$\delta_1 = \delta_2\ \delta_3\ \beta < n\ 0$
	$\delta_2 =$ neg n
	$\delta_3 =$ n

RULE	CONTROL	STACK	ENV
1	$e_0\ \gamma\ \lambda_1^n$ neg 3	e_0	e_0=PE
7	$e_0\ \gamma\ \lambda_1^n$ neg	3 e_0	
2	$e_0\ \gamma\ \lambda_1^n$	-3 e_0	
4	$e_0\ \gamma$	${}^0\lambda_1^n$ -3 e_0	
1,1	$e_0\ e_1\ \delta_2\ \delta_3\ \beta < n\ 0$	$e_1\ e_0$	$e_1=[-3/n]e_0$
6	$e_0\ e_1\ \delta_2\ \delta_3\ \beta <$	-3 0 $e_1\ e_0$	
8	$e_0\ e_1\ \delta_2\ \delta_3\ \beta$	true $e_1\ e_0$	
1,7	$e_0\ e_1$ neg n	$e_1\ e_0$	
5,5	$e_0\ e_1$	3 $e_1\ e_0$	
		3	

CSE Rules 9 and 10: Tuples

- Do not standardize "tau‟. Instead, for a tuple of the form (E1, E2, ..., En), generate the control structure tau_n E1 ... En.
- tau_n will:
  1. Pop the top n values from the stack,
  2. Create a new n-tuple,
  3. Push the tuple on the stack.

- Note: tuple elements are evaluated right-to-left.

CSE Rules 9 and 10: Tuples.

	CONTROL	STACK	ENV
CSE Rule 9 (tuple formation)	.... τ_n	V_1 ... V_n	
	....	$(V_1,...,V_n)$	
CSE Rule 10 (tuple selection)	.... γ	$(V_1,...,V_n)$ I	
	....	V_I	

CSE Rule 11: n-ary Functions

- Do not standardize the "," node.

- Instead,
  - For λ(x,y).E, simply allow multiple bindings in one environment.

CSE Rule 11: n-ary functions.

	CONTROL	STACK	ENV
CSE Rule 11 (n-ary function)	$\ldots.\ \gamma$ $\ldots.\ e_m\ \delta_k$	${}^{c}\lambda_k^{V_1,\ldots,V_n}$ Rand .... e_m	e_m=[Rand $1/V_1$]... [Rand n/V_n]e_c

Example:

Applicative expression	Control Structures
$(\lambda$(x,y).x+y) (5,6)	$\delta_0 = \gamma\ \lambda_1^{x,y}\ \tau_2$ 5 6 δ_1 = + x y

RULE	CONTROL	STACK	ENV
1,1	$e_0\ \gamma\ \lambda_1^{x,y}\ \tau_2$ 5 6	e_0	e_0=PE
9	$e_0\ \gamma\ \lambda_1^{x,y}\ \tau_2$	5 6 e_0	
2	$e_0\ \gamma\ \lambda_1^{x,y}$	(5,6) e_0	
11	$e_0\ \gamma$	${}^{0}\lambda_1^{x,y}$ (5,6) e_0	
1,1	$e_0\ e_1$ + x y	$e_1\ e_0$	e_1=[5/x][6/y]e_0
6	$e_0\ e_1$ +	5 6 $e_1\ e_0$	
5,5	$e_0\ e_1$	11 $e_1\ e_0$	
		11	

Thank You!

REFERENCES

- Programming Language Pragmatics by Michael L. Scott. 3rd edition. Morgan Kaufmann Publishers. (April 2009).
- Lecture Slides of Dr.Malaka Walpola and Dr.Bermudez