

$\{ \text{type expr} = \mid \text{Integer of int} \mid \text{Pair of expr * expr} \mid \text{Apply of string * expr}$   
 $\text{type value} = \mid \text{INT of int} \mid \text{PAIR of value * value}$

let rec eval = function

$\mid \text{Integer } n \rightarrow \text{INT } n$

$\mid \text{PAIR } (e_1, e_2) \rightarrow \text{PAIR } (\text{eval } e_1, \text{eval } e_2)$

$\mid \text{Apply } (f, e) \rightarrow \text{eval-function}(f, \text{eval } e)$

let  $v = \text{eval } e$  in  
 $\text{C}(\text{eval-function}(f, v))$

1. Add a continuation parameter  $c$  to each function, return value.

(a) let rec eval-cps  $C =$  function

$\mid \text{Integer } n \rightarrow C(\text{INT } n)$

$\mid \text{Pair } (e_1, e_2) \rightarrow \text{eval-cps } (fun v_1 \rightarrow (*PAIR 1*) \lambda v_2. C(\text{PAIR}(v_1, v_2))) e_1$   
 $\text{eval-cps } (fun v_2 \rightarrow (*PAIR 2*) \lambda v_1. C(\text{PAIR}(v_1, v_2))) e_2$

$\mid \text{Apply } (f, e) \rightarrow \text{eval-cps } (fun v \rightarrow (*FUNC*) \lambda f. C(\text{eval-function}(f, v))) e$  ..... apply-cnt

Thus

let eval-2  $e = \text{eval-cps } (fun x \rightarrow x) e$ .  $(*ID*)$  ..... part b.

eval: expr  $\rightarrow$  value.

eval-cps:  $\text{expr} \rightarrow \text{value}$

(IH)  $C(\text{eval } e) = \text{eval-cps } C e$

(b) Eliminate high-order continuations.

1. Add a constructor to  $\text{cnt}$  for each fun  $(*CNTi*) \lambda \dots$  (free variables)

type cnt =

$\mid \text{ID}$

$\mid \text{PAIR1 of expr * cnt}$

$\mid \text{PAIR2 of value * cnt}$

$\mid \text{FUNC of string * cnt}$

Call apply-cnt at every application of continuation.

let rec eval-cps-dfn  $C =$  function

$\mid \text{Integer } n \rightarrow \text{apply-cnt } C(\text{INT } n)$

$\mid \text{Pair } (e_1, e_2) \rightarrow \text{eval-cps-dfn } (*PAIR1*) (e_2, C) e_1$

$\mid \text{Apply } (f, e) \rightarrow \text{eval-cps-dfn } (*FUNC*) (f, C) e$

and apply-cnt = function

$\mid (ID, v) \rightarrow v$

$\mid (*PAIR1*) (e_2, C), v_1 \rightarrow \text{eval-cps-dfn } (*PAIR2*) (v_1, C) e_2$

$\mid (*PAIR2*) (v_1, C), v_2 \rightarrow \text{apply-cnt } (C, \text{PAIR}(v_1, v_2))$

$\mid (*FUNC*) (f, C), v \rightarrow \text{apply-cnt } (C, \text{eval-function}(f, v))$

let eval-3  $e = \text{eval-cps-dfn ID } e$ .

$fun v_1 \mapsto \text{eval-cps} \dots e_2$   
 $fun v_2 \mapsto C(\text{PAIR}(v_1, v_2))$   
 $fun v \mapsto C(\text{eval-function}(f, v))$

Mutually recursive.

eval-cps-dfn : cnt  $\rightarrow$  expr  $\rightarrow$  value

apply-cnt : cnt \* value  $\rightarrow$  value

eval-3 : expr  $\rightarrow$  value