

November 30, 2009

Programming Languages and Types Group Exercise 7

G7.1 Implementing First-Class Continuations

Consider the KCFAE-interpreter on the following page.

1. Discuss the difference between (c arg-val k) and (c arg-val) in the branch for app.
2. Explain how bindcc is implemented.

```

(define-type KCFAE-Value
  [numV (n number?)]
  [closureV (p procedure?)]
  [contV (c procedure?)])

;; interp : KCFAE Env receiver -> doesn't return
(define (interp expr env k)
  (type-case KCFAE expr
    [num (n) (k (numV n))]

    [add (l r) (interp l env (lambda (lv)
                                 (interp r env (lambda (rv)
                                               (k (num+ lv rv))))))]

    [if0 (test truth falsity)
         (interp test env (lambda (tv)
                           (if (num-zero? tv)
                               (interp truth env k)
                               (interp falsity env k))))]

    [id (v) (k (lookup v env))]

    [fun (param body)
         (k (closureV (lambda (arg-val dyn-k)
                         (interp body (aSub param arg-val env) dyn-k))))]

    [app (fun-expr arg-expr)
         (interp fun-expr env (lambda (fun-val)
                               (interp arg-expr env (lambda (arg-val)
                                         (type-case KCFAE-Value fun-val
                                           [closureV (c) (c arg-val k)]
                                           [contV (c) (c arg-val)]
                                           [else (error "not an applicable value")])))))]

    [bindcc (cont-var body)
         (interp body
                 (aSub cont-var
                       (contV (lambda (val) (k val)))
                       env)
                 k)))]
```