

EOPL Pset 4

Alireza Habibzadeh 99109393

3

الف

بدیهتا ابتدا باید تغییراتی در شکل datatype expression به وجود آوریم تا بتواند لیستی از متغیرها را در بخش `vars` و `exps` نگه دارد:

```
(define-datatype expression expression?
  (let-exp
    (vars (list-of identifier?))
    (exps (list-of expression?))
    (body expression?))
  ; .
  ; .
  ; . other expressions
)
```

تابع `value-of` هم باید تغییراتی کند:

```
(define value-of
  (λ (exp env)
    (cases expression exp
      (let-exp (vars exps body)
        (if (null? vars)
            (value-of body env)
            (value-of (let-exp ((cdr vars) (cdr exps) body))
                      (extend-env (car vars) (value-of (car exps) env) env))))
      ; .
      ; .
      ; . other expressions
    )))
```

البتّه با توجه به شکل syntax که این کار پیاده می‌شود در parser برنامه نیز تغییراتی به وجود می‌آید تا بتواند این عبارات جدید را هم parse کند.

ب

علاوه بر قسمت‌هایی که در الف تغییر دادیم باید بخش‌های مربوط به توابع بازگشتی را هم تغییر دهیم:

```
(define-datatype expression expression?
  (letrec-exp
    (p-name identifier?)
```

```

(b-vars (list-of identifier?))
(p-bodies (list-of expression?))
(letrec-body expression?))
; .
; .
; . other expressions + modified let-exp
)

```

```

(define value-of
  (λ (exp env)
    (cases expression exp
      (letrec-exp (p-name b-vars p-bodies letrec-body)
        (if (null? b-vars)
            (value-of letrec-body env)
            (value-of (letrec-exp p-name (cdr b-vars) (cdr p-bodies) letrec-body)
                      (extend-env ((car b-vars) (value-of (car p-bodies) env)) env))))
      ; .
      ; .
      ; . other expressions + modified let-exp
    )))

```