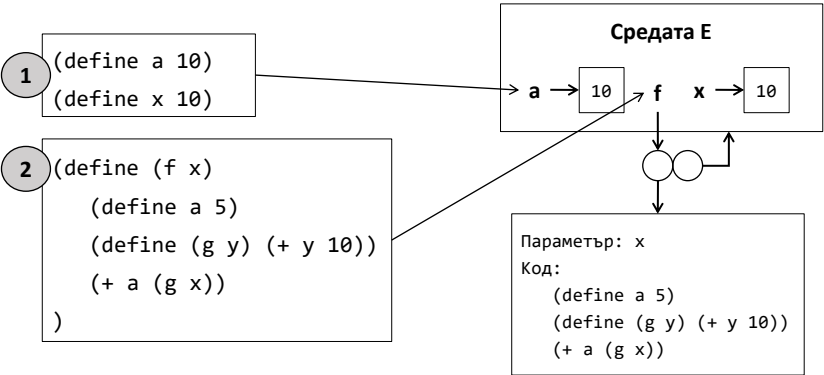


Локални дефиниции, специални форми let, let* и letrec

доц. Атанас Семерджиев

1

Локални дефиниции



2

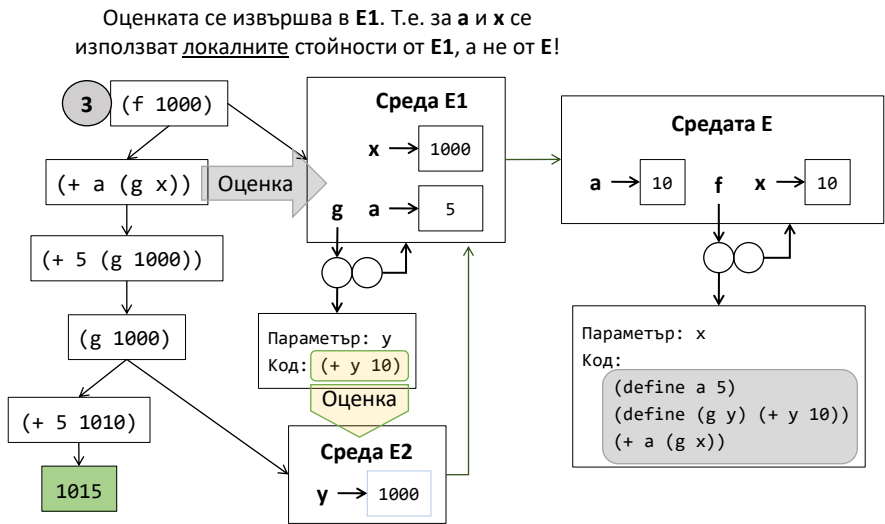
2

Локални дефиниции



3

Локални дефиниции



4

Специална форма (let)

Общ вид:

```
(let (<definitions>) <body>)
```

Позволява ни да направим локална дефиниция.

```
(let (  
    (x 5)  
    (y 10)  
    (z (+ 10 20))  
  )  
  (+ x y z)  
)
```

5

5

Специална форма (let)

- (let) не ни позволява да направим дефиниции, които зависят една от друга. Например следният код не е коректен:

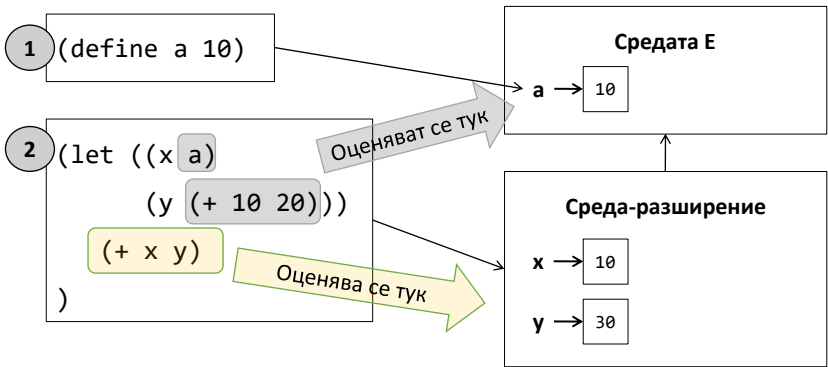
```
(let ((x 5)  
      (y (+ x 10)))  
  (+ x y)  
)
```

6

6

Семантика на (let)

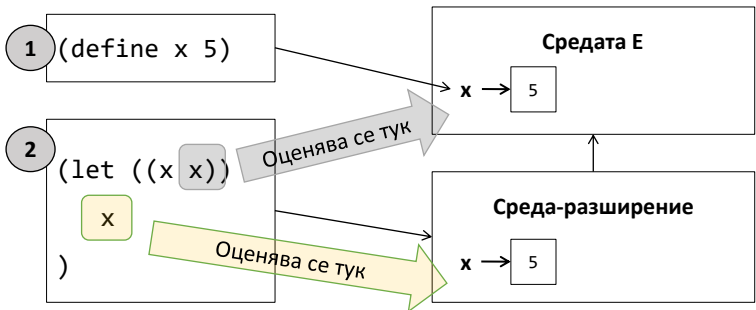
(let) създава нова среда, в която записва направените дефиниции.



7

Редът на оценяване има значение!

(let ((x x)) ...) се интерпретира така:
В средата-разширение дефинираме нов символ x, който има същата стойност, като на символа x в средата E.



8

Семантика на (let)

В средата E оценяваме:

```
(let (
  (2) (1)
  (<var1> <init1>)
  ...
  (<varn> <initn>)
)
<body> (3)
)
```

Семантика:

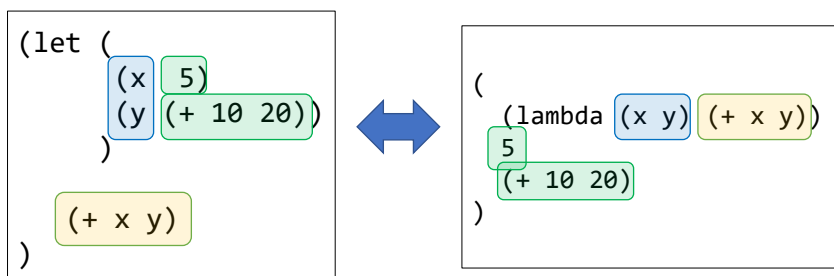
1. Създава се среда E1 -разширение на E.
2. Всеки от изразите <init_k> се оценява в средата E. Редът на оценяване е неопределен.
3. За всеки от резултатите се заделя място в паметта.
4. Символите <var_k> се добавят в средата E1 и се свързват с местата в паметта от точка 3.
5. Тялото <body> се оценява в E1.

9

9

Специална форма (let)

Важна особеност на (let) е, че тя не внася нови възможности в езика. Нейното поведение може да бъде заместено с (lambda) конструкция, например:



10

10

Специална форма (let*)

- Специалната форма (let*) има същия синтаксис като (let)
- Разликата е в това, че (let*) ни позволява да направим поредица от дефиниции, всяка от които може да използва предходните.

```
(let* ((x 5)
      (y (+ x 10)))
      (+ x y)
    )
```

11

11

Специална форма (let*)

Подобно на (let), (let*) също не внася нови възможности в езика. Тя може да се разглежда като поредица от вложени (let) конструкции:

```
(let* ((x 5)
      (y (+ x 10)))
      (+ x y)
    )
```



```
(let ((x 5))
  (let ((y (+ x 10)))
    (+ x y)
  )
)
```

12

12

Специална форма (let*)

Трябва да помним, че всяка от дефинициите в (let*) може да използва тези преди нея, но не и тези след нея!

Коректно

```
(let* ((x 5)
      (y (+ x 10)))
  (+ x y)
)
```

Некоректно

```
(let* ((y (+ x 10))
      (x 5))
  (+ x y)
)
```

13

13

Семантика на (let*)

В средата E оценяваме:

```
(let* (
  ① (<var1> <init1>)
  ...
  ② (<varn> <initn>)
)
<body> ③
)
```

Семантика:

Поведението е подобно на let, но:

- Свързванията $\langle \text{var}_k \rangle$ към $\langle \text{init}_k \rangle$ се извършват последователно, отляво-надясно.
- Всяко свързване е видимо за тези, които следват след него. Например свързването между $\langle \text{var}_1 \rangle$ и $\langle \text{init}_1 \rangle$ се извършва в среда, която е видима за $\langle \text{init}_2 \rangle$.

14

14

Специална форма (letrec)

- Има същия синтаксис като (let)
- Разликата е в това, че всяка от дефинициите в (letrec) може да използва всяка от останалите дефиниции в същия (letrec).

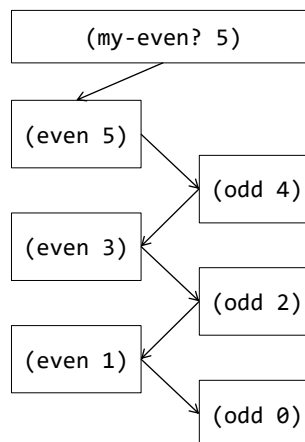
15

15

Четни числа

; Проверява дали дадено число n е четно или нечетно

```
(define (my-even? n)
  (letrec (
    (even (lambda (x)
             (if (= x 0) #t
                 (odd (- x 1)))))
    (odd (lambda (x)
           (if (= x 0) #f
               (even (- x 1)))))
  ) ; край на дефинициите
    (even n) ; тяло на letrec
  ))
```

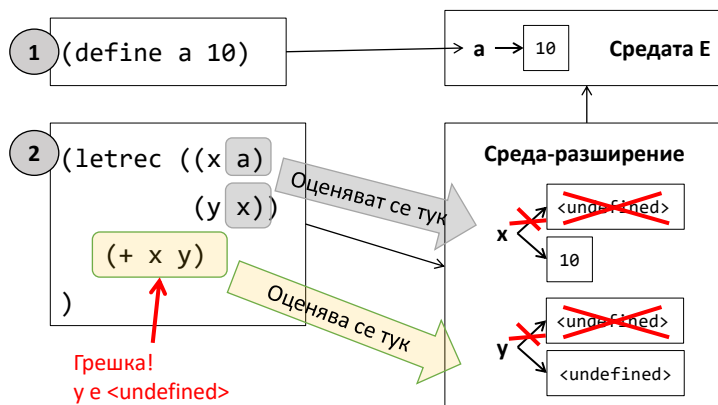


16

16

Семантика на (letrec)

По-долу е илюстрирана грешка, която може да възникне при употребата на letrec:



17

17

Семантика на (letrec)

В средата Е оценяваме:

```
(letrec ( 1 2
  (<var1> <init1>)
  ...
  (<varn> <initn>)
)
<body> 3
)
```

Семантика:

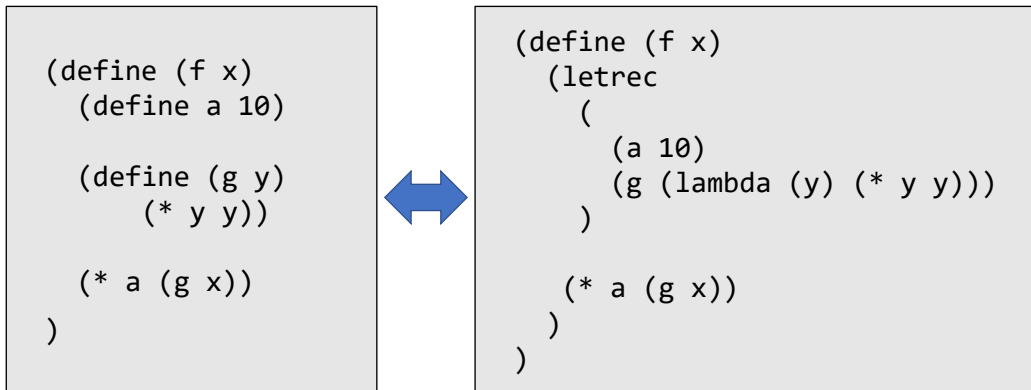
- Създава се среда E1 -разширение на E.
- Заделят се n-места, всяко от които съдържа неопределена стойност.
- Символите <var_k> се свързват с така заделените места.
- Всеки от изразите <init_k> се оценява в средата E1. Редът на оценяване е неопределен.
- Стойностите на символите <var_k> се променят на стойностите на оценените изрази <init_k>
- Тялото <body> се оценява в E1.

18

18

Специална форма (letrec)

Влагането на (define) изрази е еквивалентно на употребата на (letrec)!

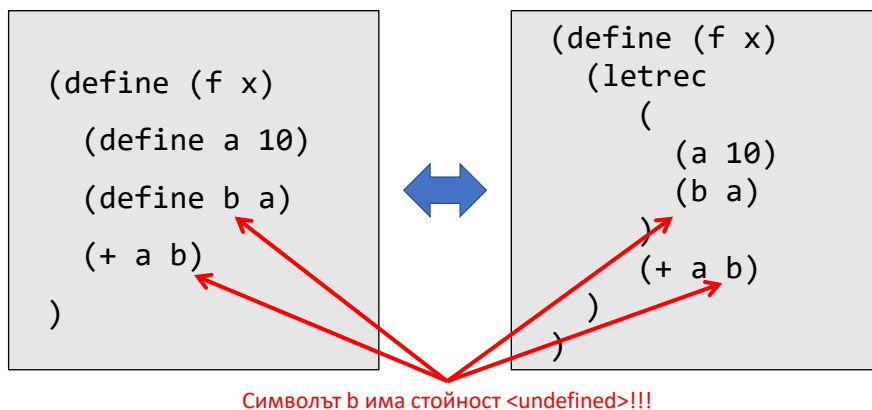


19

19

Специална форма (letrec)

Респективно, следният код е невалиден:



20

20

