Наредени двойки

доц. Атанас Семерджиев

1

Наредени двойки

Наредена двойка се създава чрез `cons`.

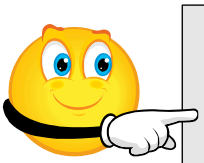
Компонентите ѝ се достъпват чрез:

- `car` – първи елемент
- `cdr` – втори елемент

```
(define p (cons 10 20))
(car p) → 10
(cdr p) → 20
```

2

2

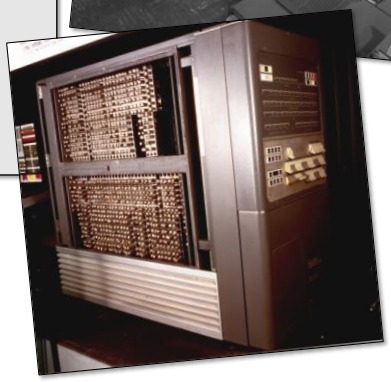


Имената на `car` и `cdr` най-вероятно идват от:

`car` → contents of the address part of register number

`cdr` → contents of the decrement part of register number

Тези понятия отдавна не са актуални, но имената на функциите са се запазили.



Източници:
https://en.wikipedia.org/wiki/CAR_and_CDR
https://en.wikipedia.org/wiki/IBM_704

3

Графично представяне

Графично ще представяме наредената двойка като двойна кутия. Например:

10	20

Интерпретаторът извежда наредените двойки като две стойности в скоби, разделени с точка:

```
(10 . 20)
```

4

4

Нотация с точка

ВНИМАНИЕ: Често срещана грешка е да се опитаме да използваме тази нотация за дефиниране на наредена двойка, т.е. да напишем

```
(10 . 20)
```

✗

вместо

```
(cons 10 20)
```

✓

Това не е коректно и интерпретаторът ще изведе съобщение за грешка.

5

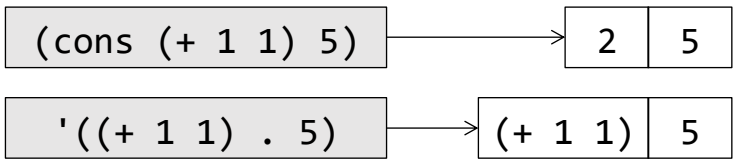
5

Нотация с точка

Въпреки казаното преди малко, има вариант да конструираме наредена двойка с помощта на символа за точка. За целта трябва да се използва специалната форма `quote`:

```
'(10 . 20)
```

Трябва обаче да внимаваме, когато използваме този подход:



6

6

Реализация с lambda

Така описаните конструкции не внасят нови възможности в езика.

Бихме могли да реализираме тази функционалност с вече известните ни конструкции.

Това е илюстрирано в следващите два примера.

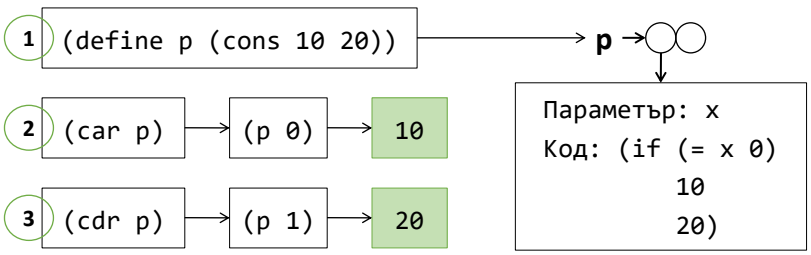
7

7

Реализация с lambda (1)

```
(define (cons a b)
   (lambda (x)
      (if (= x 0) a b)
 ))
```

```
(define (car p) (p 0))
(define (cdr p) (p 1))
```



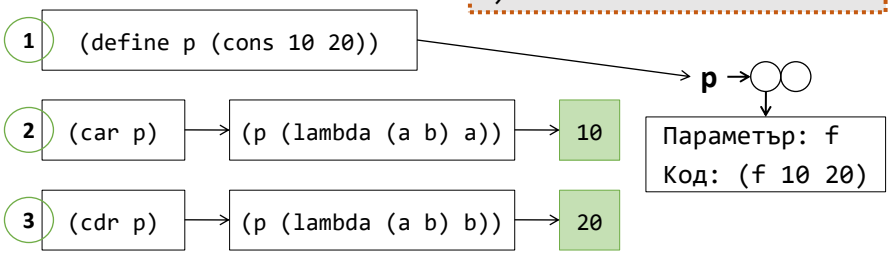
8

8

Реализация с lambda (2)

```
(define (cons a b)
   (lambda (f)
      (f a b)
 ))
```

```
(define (car p)
   (p (lambda (a b) a))
)
(define (cdr p)
   (p (lambda (a b) b))
)
```



9

9

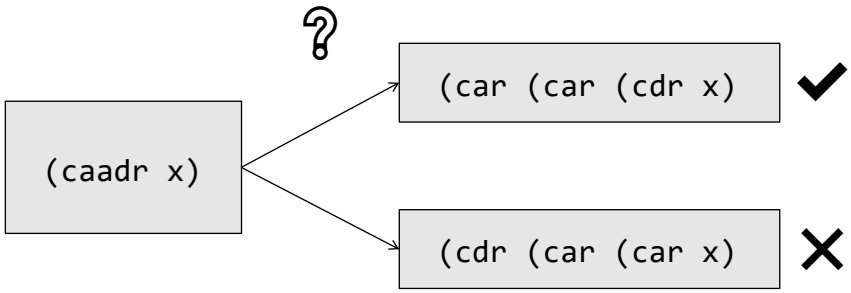
Композиция на car и cdr

```
(define x (cons (cons 1 2)
                (cons 3 4)) )
```

Израз	Оценка	Алтернативен запис
(car x)	(1 . 2)	
(cdr x)	(3 . 4)	
(car (car x))	1	(caar x)
(cdr (car x))	2	(cdar x)
(car (cdr x))	3	(cadr x)
(cdr (cdr x))	4	(cddr x)

10

Композиция на car и cdr



11

Разпознаване на наредени двойки

```
(pair? #f)     → #f
(pair? "")     → #f
(pair? "abc") → #f
(pair? 5)      → #f
(pair? (cons 1 2)) → #t
```

12

13