

Процедури от по- висок ред

23 октомври, 2018

Процедура от по-висок ред

- Процедура, която приема процедури като аргументи
- Процедура, която връща процедура
- Мощен метод за абстракция

Сума

$$\sum_{x=a}^b x$$

Сума на целите числа в интервала [a, b]

$$\sum_{x=a}^b x$$

```
(define (sum-integers a b)
  (if (> a b)
      0
      (+ a
         (sum-integers (+ a 1) b) ) ) )
```

Сума на квадратите на целите числа в интервала [a, b]

$$\sum_{x=a}^b x^2$$

```
(define (sum-squares a b)
  (if (> a b)
      0
      (+ (square a)
          (sum-squares (+ a 1) b) ) ) )
```

Сума на дроби

$$\sum_{x=a}^b \frac{1}{x^4 + 1}$$

```
(define (sum-fractions a b)
  (if (> a b)
      0
      (+ (/ 1
            (+ (square (square a)) 1))
         (sum-fractions (+ a 1) b)))))
```

Абстракция: процедура **sum**

$$\sum_{x=a}^b x$$

$$\sum_{x=a}^b x^2$$

$$\sum_{x=a}^b \frac{1}{x^4 + 1}$$

$$\sum_{x=a}^b term(x)$$

Абстракция: процедура **sum**

$$\sum_{x=a}^b term(x)$$

```
(define (sum term a b)
  (if (> a b)
    0
    (+ (term a)
      (sum term (+ a 1) b) )))
```


; Sum of all integers in [a, b] using sum

```
(define (sum-integers a b)
  (sum identity a b))
```

```
(define (identity x) x)
```

; Sum of the squares of all integers in [a, b] using sum

```
(define (sum-squares a b)
  (sum square a b))
```

```
(define (square x) (* x x))
```

; Sum of the cubes of all integers in [a, b] using sum

```
(define (sum-cubes a b)
  (sum cube a b))
```

```
(define (cube x) (* x x x))
```

; Sum of all $(1 / (x^4 + 1))$ where x is an integer in [a, b], using sum

```
(define (sum-fractions a b)
  (define (term x)
    (/ 1
      (+ (square (square x)) 1)))

  (sum term a b))
```

Произведение

$$\prod_{x=a}^b term(x)$$

```
(define (product term a b)
  (if (> a b)
    1
    (* (term a)
      (product term (+ a 1) b) )))
```

Абстракция: процедура **accumulate**

$$\sum_{x=a}^b term(x)$$

$$\prod_{x=a}^b term(x)$$

$$\bigwedge_{x=a}^b term(x)$$

...

Абстракция: процедура **accumulate**

*; Reducing the integers in [a, b] to a single value
; using the binary procedure combiner*

```
(define (accumulate combiner null-value term a b)
  (if (> a b)
    null-value
    (combiner (term a)
               (accumulate combiner
                             null-value
                             term
                             (+ a 1)
                             b) ) ) )
```

Да дефинираме **sum** и **product** чрез **accumulate**

```
(define (sum term a b)
  (accumulate + 0 term a b))
```

```
(define (product term a b)
  (accumulate * 1 term a b))
```

Дефиниране на процедури чрез **lambda**

```
(define (square x) (* x x))
```

```
(define square (lambda (x) (* x x)))
```

```
(define (square x) (* x x))
```

| | | | |
To square something, multiply it by itself.

```
(lambda (x) (+ x 4))
```

| | | |
the procedure of an argument x that adds x and 4

Използване на анонимни процедури

```
(sum 1 5 (lambda (x) (* x x))) ; 55
```

```
(sum 1 5 (lambda (x) x)) ; 15
```