

Функции от по-висок ред

Трифон Трифонов

Функционално програмиране, спец. Информатика, 2016/17 г.

20 октомври 2016 г.

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`

$$f(x) = x^2 = x$$

0,1

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → ?`

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → #t`

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → #t`
- `(fixed-point? exp 1) → ?`

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → #t`
- `(fixed-point? exp 1) → #f`

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → #t`
- `(fixed-point? exp 1) → #f`
- `(fixed-point? + 0) → ?`

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → #t`
- `(fixed-point? exp 1) → #f`
- `(fixed-point? r 0) → Грешка!`
r

(define (p x y) (+ x y))

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → #t`
- `(fixed-point? exp 1) → #f`
- `(fixed-point? + 0) → Грешка!`
- `(define (branch p? f g x) (if (p? x) (f x) (g x)))`

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → #t`
- `(fixed-point? exp 1) → #f`
- `(fixed-point? + 0) → Грешка!`
- `(define (branch p? f g x) (if (p? x) (f x) (g x)))`
- `(branch odd? exp fact 4) → ?`

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → #t`
- `(fixed-point? exp 1) → #f`
- `(fixed-point? + 0) → Грешка!`
- `(define (branch p? f g x) (if (p? x) (f x) (g x)))`
- `(branch odd? exp fact 4) → 24`

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → #t`
- `(fixed-point? exp 1) → #f`
- `(fixed-point? + 0) → Грешка!`
- `(define (branch p? f g x) (if (p? x) (f x) (g x)))`
- `(branch odd? exp fact 4) → 24`
- `(define (id x) x)`

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → #t`
- `(fixed-point? exp 1) → #f`
- `(fixed-point? + 0) → Грешка!`
- `(define (branch p? f g x) (if (p? x) (f x) (g x)))`
- `(branch odd? exp fact 4) → 24`
- `(define (id x) x)`
- `(branch number? log id "1") → ?`

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → #t`
- `(fixed-point? exp 1) → #f`
- `(fixed-point? + 0) → Грешка!`
- `(define (branch p? f g x) (if (p? x) (f x) (g x)))`
- `(branch odd? exp fact 4) → 24`
- `(define (id x) x)`
- `(branch number? log id "1") → "1"`

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → #t`
- `(fixed-point? exp 1) → #f`
- `(fixed-point? + 0) → Грешка!`
- `(define (branch p? f g x) (if (p? x) (f x) (g x)))`
- `(branch odd? exp fact 4) → 24`
- `(define (id x) x)`
- `(branch number? log id "1") → "1"`
- `(branch string? number? procedure? symbol?) → ?`

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → #t`
- `(fixed-point? exp 1) → #f`
- `(fixed-point? + 0) → Грешка!`
- `(define (branch p? f g x) (if (p? x) (f x) (g x)))`
- `(branch odd? exp fact 4) → 24`
- `(define (id x) x)`
- `(branch number? log id "1") → "1"`
- `(branch string? number? procedure? symbol?) → #t`

Подаване на функции като параметри

В Scheme функциите са “първокласни” стойности.

Примери:

- `(define (fixed-point? f x) (= (f x) x))`
- `(fixed-point? sin 0) → #t`
- `(fixed-point? exp 1) → #f`
- `(fixed-point? + 0) → Грешка!`
- `(define (branch p? f g x) ((if (p? x) f g) x))`
- `(branch odd? exp fact 4) → 24`
- `(define (id x) x)`
- `(branch number? log id "1") → "1"`
- `(branch string? number? procedure? symbol?) → #t`

Функции от по-висок ред

Дефиниция

Функция, която приема функция за параметър се нарича *функция от по-висок ред*.

Функции от по-висок ред

Дефиниция

Функция, която приема функция за параметър се нарича *функция от по-висок ред*.

- `fixed-point?` и `branch` са функции от по-висок ред

Функции от по-висок ред

Дефиниция

Функция, която приема функция за параметър се нарича *функция от по-висок ред*.

- fixed-point? и branch са функции от по-висок ред
- Примери за математически функции от по-висок ред?

$$\begin{aligned}
 &': (\mathbb{R} \rightarrow \mathbb{R}) \rightarrow (\mathbb{R} \rightarrow \mathbb{R}) \\
 &\int dx : (\mathbb{R} \rightarrow \mathbb{R}) \rightarrow (\mathbb{R} \rightarrow (\mathbb{R} \rightarrow \mathbb{R})) \\
 &\int_6 dx: \mathbb{R} \rightarrow \mathbb{R} \rightarrow (\mathbb{R} \rightarrow \mathbb{R}) \rightarrow \mathbb{R}
 \end{aligned}$$

Функции от по-висок ред

Дефиниция

Функция, която приема функция за параметър се нарича *функция от по-висок ред*.

- `fixed-point?` и `branch` са функции от по-висок ред
- Примери за математически функции от по-висок ред?
- Всички функции в λ -смятането са от по-висок ред!

Задачи за сумиране

Задача: Да се пресметнат следните суми:

① $k^2 + (k + 1)^2 + \dots + 100^2$ за $k \leq 100$

② $\int_a^b f(x) \approx \Delta x [f(a) + f(a + \Delta x) + f(a + 2\Delta x) + \dots + f(b)]$

③ $x + e^x + e^{e^x} + e^{e^{e^x}} + \dots$ докато поредното събираемо е $\leq 10^{1000}$



Задачи за сумиране

Задача: Да се пресметнат следните суми:

- ① $k^2 + (k + 1)^2 + \dots + 100^2$ за $k \leq 100$
- ② $\int_a^b f(x) \approx \Delta x [f(a) + f(a + \Delta x) + f(a + 2\Delta x) + \dots + f(b)]$
- ③ $x + e^x + e^{e^x} + e^{e^{e^x}} + \dots$ докато поредното събираемо е $\leq 10^{1000}$

```
(define (sum1 k)
  (if (> k 100) 0 (+ (* k k) (sum1 (+ k 1)))))
```


Задачи за сумиране

Задача: Да се пресметнат следните суми:

① $k^2 + (k + 1)^2 + \dots + 100^2$ за $k \leq 100$

② $\int_a^b f(x) \approx \Delta x [f(a) + f(a + \Delta x) + f(a + 2\Delta x) + \dots + f(b)]$

③ $x + e^x + e^{e^x} + e^{e^{e^x}} + \dots$ докато поредното събираемо е $\leq 10^{1000}$

$$\int_a^b f(x) dx = \int_a^{a+\Delta x} f(x) dx + \int_{a+\Delta x}^b f(x) dx$$

```
(define (sum1 k)
```

```
  (if (> k 100) 0 (+ (* k k) (sum1 (+ k 1)))))
```

```
(define (sum2 a b f dx)
```

```
  (if (> a b) 0 (+ (* dx (f a)) (sum2 (+ a dx) b f dx))))
```

Задачи за сумиране

Задача: Да се пресметнат следните суми:

① $k^2 + (k + 1)^2 + \dots + 100^2$ за $k \leq 100$

② $\int_a^b f(x) \approx \Delta x [f(a) + f(a + \Delta x) + f(a + 2\Delta x) + \dots + f(b)]$

③ $x + e^x + e^{e^x} + e^{e^{e^x}} + \dots$ докато поредното събираемо $e \leq 10^{1000}$

`(define (sum1 k)`
`(if (> k 100) 0 (+ (* k k) (sum1 (+ k 1))))))`

y := e^x
x + y + e^y + e^e^y

`(define (sum2 a b f dx)`
`(if (> a b) 0 (+ (* dx (f a)) (sum2 (+ a dx) b f dx))))`

`(define (sum3 x)`
`(if (> x (expt 10 1000)) 0 (+ x (sum3 (exp x)))))`

y

Задачи за сумиране

Задача: Да се пресметнат следните суми:

- ① $k^2 + (k + 1)^2 + \dots + 100^2$ за $k \leq 100$
- ② $\int_a^b f(x) \approx \Delta x [f(a) + f(a + \Delta x) + f(a + 2\Delta x) + \dots + f(b)]$
- ③ $x + e^x + e^{e^x} + e^{e^{e^x}} + \dots$ докато поредното събираемо е $\leq 10^{1000}$

```
(define (sum1 k)
  (if (> k 100) 0 (+ (* k k) (sum1 (+ k 1)))))

(define (sum2 a b f dx)
  (if (> a b) 0 (+ (* dx (f a)) (sum2 (+ a dx) b f dx))))

(define (sum3 x)
  (if (> x (exp 10 1000)) 0 (+ x (sum3 (exp x)))))
```

Задачи за сумиране

Задача: Да се пресметнат следните суми:

- ① $k^2 + (k + 1)^2 + \dots + 100^2$ за $k \leq 100$
- ② $\int_a^b f(x) \approx \Delta x [f(a) + f(a + \Delta x) + f(a + 2\Delta x) + \dots + f(b)]$
- ③ $x + e^x + e^{e^x} + e^{e^{e^x}} + \dots$ докато поредното събираемо е $\leq 10^{1000}$

```
(define (sum1 k)
  (if (> k 100) 0 (+ (* k k) (sum1 (+ k 1)))))

(define (sum2 a b f dx)
  (if (> a b) 0 (+ (* dx (f a)) (sum2 (+ a dx) b f dx))))

(define (sum3 x)
  (if (> x (exp 10 1000)) 0 (+ x (sum3 (exp x)))))
```

Задачи за сумиране

Задача: Да се пресметнат следните суми:

- ① $k^2 + (k + 1)^2 + \dots + 100^2$ за $k \leq 100$
- ② $\int_a^b f(x) \approx \Delta x [f(a) + f(a + \Delta x) + f(a + 2\Delta x) + \dots + f(b)]$
- ③ $x + e^x + e^{e^x} + e^{e^{e^x}} + \dots$ докато поредното събираемо е $\leq 10^{1000}$

```
(define (sum1 k)
  (if (> k 100) 0 (+ (* k k) (sum1 (+ k 1)))))

(define (sum2 a b f dx)
  (if (> a b) 0 (+ (* dx (f a)) (sum2 (+ a dx) b f dx))))

(define (sum3 x)
  (if (> x (exp 10 1000)) 0 (+ x (sum3 (exp x))))))
```

Обобщена функция за сумиране

Да се напише функция от по-висок ред `sum`, която пресмята сумата:

$$\sum_{\substack{i=a \\ i \leftarrow \text{next}(i)}}^b \text{term}(i).$$

Обобщена функция за сумиране

Да се напише функция от по-висок ред `sum`, която пресмята сумата:

$$\sum_{\substack{i=a \\ i \leftarrow \text{next}(i)}}^b \text{term}(i).$$

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

Приложения на sum

Решение на задачите за суми чрез sum:

$$\sum_{i=k}^{100} i^2$$

Приложения на sum

Решение на задачите за суми чрез sum:

$$\sum_{i=k}^{100} i^2$$

```
(define (square x) (* x x))  
(define (1+ x) (+ x 1))  
(define (sum1 k) (sum k 100 square 1+))
```

Приложения на sum

Решение на задачите за суми чрез sum:

$$\sum_{i=k}^{100} i^2$$

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

```
(define (1+ x) (+ x 1))
```

```
(define (sum1 k) (sum k 100 square 1+))
```

$$\sum_{\substack{i=a \\ i \rightarrow i+\Delta x}}^b \Delta x f(i)$$

Приложения на sum

Решение на задачите за суми чрез sum:

$$\sum_{i=k}^{100} i^2$$

```
(define (square x) (* x x))
(define (1+ x) (+ x 1))
(define (sum1 k) (sum k 100 square 1+))
```

$$\sum_{\substack{i=a \\ i \rightarrow i+\Delta x}}^b \Delta x f(i)$$

```
(define (sum2 a b f dx)
  (define (term x) (* dx (f x)))
  (define (next x) (+ x dx))
  (sum a b term next))
```

Приложения на sum

Решение на задачите за суми чрез sum:

$$\sum_{i=k}^{100} i^2$$

```
(define (square x) (* x x))
(define (1+ x) (+ x 1))
(define (sum1 k) (sum k 100 square 1+))
```

$$\sum_{\substack{i=a \\ i \rightarrow i+\Delta x}}^b \Delta x f(i)$$

```
(define (sum2 a b f dx)
  (define (next x) (+ x dx))
  (* dx (sum a b f next)))
```

Приложения на sum

Решение на задачите за суми чрез sum:

$$\sum_{i=k}^{100} i^2$$

```
(define (square x) (* x x))
(define (1+ x) (+ x 1))
(define (sum1 k) (sum k 100 square 1+))
```

$$\sum_{\substack{i=a \\ i \rightarrow i+\Delta x}}^b \Delta x f(i)$$

```
(define (sum2 a b f dx)
  (define (next x) (+ x dx))
  (* dx (sum a b f next)))
```

$$\sum_{\substack{i=x \\ i \rightarrow e^i}}^{10^{1000}} i$$

Приложения на sum

Решение на задачите за суми чрез sum:

$$\sum_{i=k}^{100} i^2$$

```
(define (square x) (* x x))
(define (1+ x) (+ x 1))
(define (sum1 k) (sum k 100 square 1+))
```

$$\sum_{\substack{i=a \\ i \rightarrow i+\Delta x}}^b \Delta x f(i)$$

```
(define (sum2 a b f dx)
  (define (next x) (+ x dx))
  (* dx (sum a b f next)))
```

$$\sum_{\substack{i=x \\ i \rightarrow e^i}}^{10^{1000}} i$$

```
(define (sum3 x)
  (sum x (expt 10 1000) id exp))
```

Обобщена функция за произведение

Да се напише функция от по-висок ред `product`, която пресмята:

$$\prod_{\substack{i=a \\ i \leftarrow \text{next}(i)}}^b \text{term}(i).$$

Обобщена функция за произведение

Да се напише функция от по-висок ред `product`, която пресмята:

$$\prod_{\substack{i=a \\ i \leftarrow \text{next}(i)}}^b \text{term}(i).$$

```
(define (prod a b term next)
  (if (> a b) 1 (* (term a) (prod (next a) b term next))))
```


Обобщена функция за произведение

Да се напише функция от по-висок ред `product`, която пресмята:

$$\prod_{\substack{i=a \\ i \leftarrow \text{next}(i)}}^b \text{term}(i).$$

```
(define (prod a b term next)
  (if (> a b) 1 (* (term a) (prod (next a) b term next))))

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

Обобщена функция за произведение

Да се напише функция от по-висок ред `product`, която пресмята:

$$\prod_{\substack{i=a \\ i \leftarrow \text{next}(i)}}^b \text{term}(i).$$

```
(define (prod a b term next)
  (if (> a b) 1 (* (term a) (prod (next a) b term next))))
```

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

Обобщена функция за натрупване

Да се напише функция, която пресмята

$$term(a) \oplus \left(term(next(a)) \oplus \left(\dots \oplus (term(b) \oplus \perp) \dots \right) \right),$$

където \oplus е бинарна операция,
а \perp е нейната “нулева стойност”, т.е. $x \oplus \perp = x$.

Обобщена функция за натрупване

Да се напише функция, която пресмята

$$term(a) \oplus \left(term(next(a)) \oplus \left(\dots \oplus (term(b) \oplus \perp) \dots \right) \right),$$

където \oplus е бинарна операция,
а \perp е нейната “нулева стойност”, т.е. $x \oplus \perp = x$.

```
(define (accumulate op nv a b term next)
  (if (> a b) nv
      (op (term a) (accumulate op nv (next a) b term next))))
```

Обобщена функция за натрупване

Да се напише функция, която пресмята

$$term(a) \oplus \left(term(next(a)) \oplus \left(\dots \oplus (term(b) \oplus \perp) \dots \right) \right),$$

където \oplus е бинарна операция,

$a \perp$ е нейната “нулева стойност”, т.е. $x \oplus \perp = x$.

```
(define (accumulate op nv a b term next)
  (if (> a b) nv
      (op (term a) (accumulate op nv (next a) b term next))))

(define (sum a b term next) (accumulate + 0 a b term next))
(define (product a b term next) (accumulate * 1 a b term next))
```