

Функции от по-висок ред

Трифон Трифонов

Функционално програмиране, 2017/18 г.

26 октомври 2017 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))`

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? expt 0) → ?`

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? expt 0) → Грешка!`

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? expt 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?` `expt` 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?` `expt` 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? expt 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? expt 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? expt 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? expt 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? expt 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? expt 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

$$\frac{\partial}{\partial r} : (\mathbb{R} \rightarrow \mathbb{R}) \rightarrow (\mathbb{R} \rightarrow \mathbb{R}) \quad \int dx : \mathbb{R} \rightarrow \mathbb{R} \rightarrow (\mathbb{R} \rightarrow \mathbb{R}) \rightarrow \mathbb{R}$$

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?

$$\int dx : (\mathbb{R} \rightarrow \mathbb{R}) \rightarrow (\mathbb{R} \rightarrow (\mathbb{R} \rightarrow \mathbb{R})) \int x^2 dx = \frac{x^3}{3} + C$$

$$h_a n^b : \mathbb{N} \rightarrow \mathbb{R}$$

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 $\int_a^b f(x) dx \approx \Delta x [f(a) + f(a + \Delta x) + f(a + 2\Delta x) + \dots + f(b)]$
- ③ $x + e^x + e^{e^x} + e^{e^{e^x}} + \dots$ докато поредното събираме $e \leq 10^{1000}$



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 събираме $e \leq 10^{1000}$

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

Задачи за сумиране

Задача: Да се пресметнат следните суми:

$$\textcircled{1} \quad k^2 + (k+1)^2 + \dots + 100^2 \text{ за } k \leq 100$$

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

$$\textcircled{3} \quad x + e^x + e^{e^x} + e^{e^{e^x}} + \dots \text{ докато поредното събираме } e \leq 10^{1000}$$

```
(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))))
```



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

Задачи за сумиране

Задача: Да се пресметнат следните суми:

$$\textcircled{1} \quad k^2 + (k+1)^2 + \dots + 100^2 \text{ за } k \leq 100$$

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

$$\textcircled{3} \quad x + e^x + e^{e^x} + e^{e^{e^x}} + \dots \text{ докато поредното събираме } e \leq 10^{1000}$$

(define (sum1 k) *y := e^x* *x + y + e^y + ...*)

```
(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 (expt 10 1000)) 0 (+ x (sum3 (exp x)))))
```

Задачи за сумиране

Задача: Да се пресметнат следните суми:

$$\textcircled{1} \quad k^2 + (k+1)^2 + \dots + 100^2 \text{ за } k \leq 100$$

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

$$\textcircled{3} \quad x + e^x + e^{e^x} + e^{e^{e^x}} + \dots \text{ докато поредното събираме } e \leq 10^{1000}$$

```
(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 (expt 10 1000)) 0 (+ x (sum3 (exp x)))))
```

Задачи за сумиране

Задача: Да се пресметнат следните суми:

$$\textcircled{1} \quad k^2 + (k+1)^2 + \dots + 100^2 \text{ за } k \leq 100$$

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

$$\textcircled{3} \quad x + e^x + e^{e^x} + e^{e^{e^x}} + \dots \text{ докато поредното събираме } e \leq 10^{1000}$$

```
(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 (expt 10 1000)) 0 (+ x (sum3 (exp x)))))
```

Задачи за сумиране

Задача: Да се пресметнат следните суми:

$$\textcircled{1} \quad k^2 + (k+1)^2 + \dots + 100^2 \text{ за } k \leq 100$$

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

$$\textcircled{3} \quad x + e^x + e^{e^x} + e^{e^{e^x}} + \dots \text{ докато поредното събираме } e \leq 10^{1000}$$

```
(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 (expt 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+))
```

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

(Sum a b term next)
 ↓ ↓ i → Δx f(i) Δx
 a b

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+))
```

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

```
(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+))
```

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

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

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

(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+))
```

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

```
(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:

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

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

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

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

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

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

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

където \oplus е бинарна операция,

а \perp е нейната "нулева стойност", т.e. $x \oplus \perp = x$.

```
(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))
```

Задача: пресмятане на полином

Да се пресметне стойността на полинома

$$P_n(x) = x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1)$$

Задача: пресмятане на полином

Да се пресметне стойността на полинома

$$\begin{aligned}P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\&= \sum_{i=0}^n (n+1-i)x^i\end{aligned}$$

Задача: пресмятане на полином

Да се пресметне стойността на полинома

$$\begin{aligned}P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\&= \sum_{i=0}^n (n+1-i)x^i\end{aligned}$$

Решение №1:

```
(define (p n x)
```

```
(accumulate + ? ? ? ? ?))
```

Задача: пресмятане на полином

Да се пресметне стойността на полинома

$$\begin{aligned}P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\&= \sum_{i=0}^n (n+1-i)x^i\end{aligned}$$

Решение №1:

```
(define (p n x)
  (accumulate + 0 ? ? ? ?))
```

Задача: пресмятане на полином

Да се пресметне стойността на полинома

$$\begin{aligned}P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\&= \sum_{i=0}^n (n+1-i)x^i\end{aligned}$$

Решение №1:

```
(define (p n x)
```

```
(accumulate + 0 0 ? ? ?))
```

Задача: пресмятане на полином

Да се пресметне стойността на полинома

$$\begin{aligned}P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\&= \sum_{i=0}^n (n+1-i)x^i\end{aligned}$$

Решение №1:

```
(define (p n x)
```

```
(accumulate + 0 0 n ? ?))
```

Задача: пресмятане на полином

Да се пресметне стойността на полинома

$$\begin{aligned}P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\&= \sum_{i=0}^n (n+1-i)x^i\end{aligned}$$

Решение №1:

```
(define (p n x)
  (define (term i) (* (- (1+ n) i) (expt x i)))
  (accumulate + 0 0 n term ?))
```

Задача: пресмятане на полином

Да се пресметне стойността на полинома

$$\begin{aligned}P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\&= \sum_{i=0}^n (n+1-i)x^i\end{aligned}$$

Решение №1:

```
(define (p n x)
  (define (term i) (* (- (1+ n) i) (expt x i)))
  (accumulate + 0 0 n term 1+))
```

Задача: пресмятане на полином

Да се пресметне стойността на полинома

$$\begin{aligned}P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\&= \sum_{i=0}^n (n+1-i)x^i\end{aligned}$$

Решение №1:

```
(define (p n x)
  (define (term i) (* (- (1+ n) i) (expt x i)))
  (accumulate + 0 0 n term 1+))
```

Можем ли да решим задачата без да извикваме expt на всяка стъпка?

Правило на Хорнер

$$\begin{aligned}P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\&= \left(\left(\left(\dots \left((x+2)x + 3 \right) x + \dots \right) x + (n-1) \right) x + n \right) x + (n+1)\end{aligned}$$

Правило на Хорнер

$$\begin{aligned}P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\&= \left(\left(\left(\dots ((x+2)x+3)x+\dots \right)x+(n-1) \right)x+n \right)x+(n+1)\end{aligned}$$

Можем ли да сметнем с accumulate?

Правило на Хорнер

$$\begin{aligned}P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\&= \left(\left(\left(\dots ((x+2)x+3)x+\dots \right)x+(n-1) \right)x+n \right)x+(n+1)\end{aligned}$$

Можем ли да сметнем с accumulate?

Идея: Да използваме операцията $a \oplus b := ax + b$.

Правило на Хорнер

$$\begin{aligned}
 P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\
 &= \left(\left(\left(\dots ((x+2)x+3)x+\dots \right)x+(n-1) \right)x+n \right)x+(n+1)
 \end{aligned}$$

Можем ли да сметнем с accumulate?

Идея: Да използваме операцията $a \oplus b := ax + b$.

Коя е "нулевата стойност" \perp ?

$$\begin{aligned}
 a \times \perp &= \perp \\
 \perp \oplus b &= b
 \end{aligned}$$

Правило на Хорнер

$$\begin{aligned}
 P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\
 &= \left(\left(\left(\dots ((x+2)x+3)x+\dots \right)x+(n-1) \right)x+n \right)x+(n+1)
 \end{aligned}$$

Можем ли да сметнем с accumulate?

Идея: Да използваме операцията $a \oplus b := ax + b$.

Коя е “нулевата стойност” \perp ?

Решение №2:

```
(define (p n x)
  (define (op a b) (+ (* a x) b))
  (accumulate op 0 1 (1+ n) id 1+))
```

Правило на Хорнер

$$\begin{aligned}
 P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\
 &= \left(\left(\left(\dots ((x+2)x+3)x+\dots \right)x+(n-1) \right)x+n \right)x+(n+1)
 \end{aligned}$$

Можем ли да сметнем с accumulate?

Идея: Да използваме операцията $a \oplus b := ax + b$.

Коя е “нулевата стойност” \perp ?

Решение №2:

```
(define (p n x)
  (define (op a b) (+ (* a x) b))
  (accumulate op 0 1 (1+ n) id 1+))
```

Не смята правилно!

Правило на Хорнер

Всъщност пресметнахме:

$$Q_n(x) = x + 2x + 3x + \dots + nx + (n+1)x = \frac{(n+1)(n+2)}{2}x.$$

$$\textcircled{4} := ax+b \quad 1.x + (2.x + (-\dots\dots))$$

form := id

a := 1

Правило на Хорнер

Всъщност пресметнахме:

$$Q_n(x) = x + 2x + 3x + \dots + nx + (n+1)x = \frac{(n+1)(n+2)}{2}x.$$

Идея: Да използваме операцията $a \oplus b := a + bx$.

Правило на Хорнер

Всъщност пресметнахме:

$$Q_n(x) = x + 2x + 3x + \dots + nx + (n+1)x = \frac{(n+1)(n+2)}{2}x.$$

Идея: Да използваме операцията $a \oplus b := a + bx$.

Решение №3:

```
(define (p n x)
  (define (op a b) (+ a (* b x)))
  (accumulate op 0 1 (1+ n) id 1+))
```

Правило на Хорнер

Всъщност пресметнахме:

$$Q_n(x) = x + 2x + 3x + \dots + nx + (n+1)x = \frac{(n+1)(n+2)}{2}x.$$

Идея: Да използваме операцията $a \oplus b := a + bx$.

Решение №3:

```
(define (p n x)
  (define (op a b) (+ a (* b x)))
  (accumulate op 0 1 (1+ n) id 1+))
```

1 + (2 + (1 + (.....)) * x)

Пак не смята правилно!!!

Ляво и дясно натрупване

Всъщност пресметнахме:

$$\begin{aligned} R_n(x) &= 1 + x \left(2 + x \left(\dots + x \left((n-1) + x(n+x(n+1)) \right) \dots \right) \right) \\ &= (n+1)x^n + nx^{n-1} + (n-1)x^{n-2} + \dots + 3x^2 + 2x + 1 \end{aligned}$$

вместо

$$\begin{aligned} P_n(x) &= \left(\left(\left(\dots ((x+2)x+3)x+\dots \right) x+(n-1) \right) x+n \right) x+(n+1) \\ &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1). \end{aligned}$$

Ляво и дясно натрупване

Всъщност пресметнахме:

$$(5/3)/2 \neq 5/(3/2)$$

$$\begin{aligned} R_n(x) &= 1 + x \left(2 + x \left(\dots + x \left((n-1) + x(n+x(n+1)) \right) \dots \right) \right) \\ &= (n+1)x^n + nx^{n-1} + (n-1)x^{n-2} + \dots + 3x^2 + 2x + 1 \end{aligned}$$

вместо

$$\begin{aligned} P_n(x) &= \left(\left(\left(\dots ((x+2)x+3)x + \dots \right) x + (n-1) \right) x + n \right) x + (n+1) \\ &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1). \end{aligned}$$

За неассоциативни операции \oplus има значение в какъв ред са скобите!

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C$$

Обобщена функция за ляво натрупване

Да се напише функция, която пресмята ляво натрупване:

$$\left(\dots \left(\underbrace{(\perp \oplus term(a)) \oplus term(next(a))}_{\perp} \right) \oplus \dots \right) \oplus term(b)$$

\vdash

$$(\perp \oplus term(u)) , \vdash \dots \vdash \perp \oplus term(u)$$

Обобщена функция за ляво натрупване

Да се напише функция, която пресмята **ляво натрупване**:

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

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

Обобщена функция за ляво натрупване

Да се напише функция, която пресмята **ляво натрупване**:

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

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

- `accumulate` — дясно натрупване, рекурсивен процес
- `accumulate-i` — ляво натрупване, итеративен процес

Правило на Хорнер

$$\begin{aligned}P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\&= \left(\left(\left(\dots \left((x+2)x + 3 \right) x + \dots \right) x + (n-1) \right) x + n \right) x + (n+1)\end{aligned}$$

Идея: използваме `accumulate`-и и $a \oplus b := ax + b$.

Правило на Хорнер

$$\begin{aligned}P_n(x) &= x^n + 2x^{n-1} + \dots + (n-2)x^3 + (n-1)x^2 + nx + (n+1) \\&= \left(\left(\left(\dots ((x+2)x+3)x+\dots \right)x+(n-1) \right)x+n \right)x+(n+1)\end{aligned}$$

Идея: използваме accumulate-и и $a \oplus b := ax + b$.

Решение №4:

```
(define (p n x)
  (define (op a b) (+ (* a x) b))
  (accumulate-i op 0 1 (1+ n) id 1+))
```

Анонимни функции

Можем ли да ги конструираме параметрите на функциите от по-висок ред "на място", без да даваме имена?

Анонимни функции

Можем ли да ги конструираме параметрите на функциите от по-висок ред "на място", без да даваме имена?

- `(lambda (<параметър>) <тяло>)`

Анонимни функции

Можем ли да ги конструираме параметрите на функциите от по-висок ред "на място", без да даваме имена?

- `(lambda ({<параметър>}) <тяло>)`
- Оценява се до функционален обект със съответните параметри и тяло

Анонимни функции

Можем ли да ги конструираме параметрите на функциите от по-висок ред "на място", без да даваме имена?

- `(lambda ({<параметър>}) <тяло>)`
- Оценява се до функционален обект със съответните параметри и тяло
- Анонимната функция пази указател към средата, в която е оценена

Анонимни функции

Можем ли да ги конструираме параметрите на функциите от по-висок ред "на място", без да даваме имена?

- `(lambda ({<параметър>}) <тяло>)`
- Оценява се до функционален обект със съответните параметри и тяло
- Анонимната функция пази указател към средата, в която е оценена
- Примери:

Анонимни функции

Можем ли да ги конструираме параметрите на функциите от по-висок ред "на място", без да даваме имена?

- **(lambda ({<параметър>}) <тяло>)**
- Оценява се до функционален обект със съответните параметри и тяло
- Анонимната функция пази указател към средата, в която е оценена
- Примери:
 - `(lambda (x) (+ x 3))` → #<procedure>

Анонимни функции

Можем ли да ги конструираме параметрите на функциите от по-висок ред "на място", без да даваме имена?

- **(lambda ({<параметър>}) <тяло>)**
- Оценява се до функционален обект със съответните параметри и тяло
- Анонимната функция пази указател към средата, в която е оценена
- Примери:
 - **(lambda (x) (+ x 3))** → #<procedure>
 - **((lambda (x) (+ x 3)) 5)** → 8

Анонимни функции

Можем ли да ги конструираме параметрите на функциите от по-висок ред "на място", без да даваме имена?

- **(lambda (<параметър>) <тяло>)**
- Оценява се до функционален обект със съответните параметри и тяло
- Анонимната функция пази указател към средата, в която е оценена
- Примери:
 - `(lambda (x) (+ x 3))` → #<procedure>
 - `((lambda (x) (+ x 3)) 5)` → 8
 - `(define (<име> <параметри>) <тяло>)`
↔
`(define <име> (lambda (<параметри>) <тяло>))`

Примери

```
(define (integral a b f dx)
  (* dx (accumulate + 0 a b f (lambda (x) (+ x dx)))))
```

Примери

```
(define (integral a b f dx)
  (* dx (accumulate + 0 a b f (lambda (x) (+ x dx)))))

(define (p n x)
  (accumulate-i + 0 1 (+ n 1) (lambda (a b) (+ (* a x) b))
                (lambda (i) (+ i 1))))
```

Примери

```
(define (integral a b f dx)
  (* dx (accumulate + 0 a b f (lambda (x) (+ x dx)))))

(define (p n x)
  (accumulate-i + 0 1 (+ n 1) (lambda (a b) (+ (* a x) b))
                (lambda (i) (+ i 1))))
```

Задача: Как можем да реализираме ~~accumulate~~:

- $n!$

$$1 \cdot 2 \cdot 3 \cdots h$$

$$\prod_{i=1}^n ?$$

Примери

```
(define (integral a b f dx)
  (* dx (accumulate + 0 a b f (lambda (x) (+ x dx)))))

(define (p n x)
  (accumulate-i + 0 1 (+ n 1) (lambda (a b) (+ (* a x) b))
                (lambda (i) (+ i 1))))
```

Задача: Как можем да реализираме с accumulate:

- $n!$
- x^n

X V, ... X

Примери

```
(define (integral a b f dx)
  (* dx (accumulate + 0 a b f (lambda (x) (+ x dx)))))

(define (p n x)
  (accumulate-i + 0 1 (+ n 1) (lambda (a b) (+ (* a x) b))
                (lambda (i) (+ i 1))))
```

Задача: Как можем да реализираме с accumulate:

- $n!$

$$\left((0+1) \cdot \frac{1}{1} + (1+1) \cdot \frac{1}{2} + \dots + (n+1) \cdot \frac{1}{n+1} \right) \cdot x + 1$$

- x^n

$$\sum_{i=0}^n \frac{x^i}{i!} \approx e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$$

$$= 1 + \frac{x}{1} \left(1 + \frac{x}{2} \left(1 + \frac{x}{3} \left(1 + \dots + \frac{x}{n} (1+0) \right) \right) \right)$$