

Paradigma de Programação Funcional

Elias Italiano Rodrigues – 7987251

Vinicius Katata Biondo – 6783972

SCC0217 – Linguagens de Programação e Compiladores

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

5 de julho de 2015

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

Tópicos

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

Tópicos

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

O Paradigma

- O paradigma funcional é baseado em funções matemáticas.
- Foi idealizado com o objetivo de criar programas mais legíveis, com funções independentes de contexto, sem efeitos colaterais e mudança de estado durante a execução do programa.
- LISP é uma das linguagens mais conhecidas.

Tópicos

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

- Uma função matemática é o mapeamento de elementos de um conjunto em outro conjunto.
- As expressões matemáticas são expressas por condições e controladas por recursão.
- Não dependem de valores externos.
- O valor de um parâmetro não é alterado durante a avaliação da função.

Tópicos

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

- Expressões Lambda

É uma notação para escrever funções sem fazer referência a um nome.

$$\lambda(x) \ x * x * x$$

- *Lambda Calculus*

É um sistema para definição e aplicação de funções e recursão usando expressões lambda.

Tópicos

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

- Objetivo do paradigma funcional é imitar funções matemáticas; resulta numa abordagem diferente de resolução de problemas.
- Não existem o controle de memória, atribuição de dados e estrutura de repetição.
- As linguagens funcionais possuem um conjunto primitivo de funções.

Tópicos

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

Atualmente muitas linguagens funcionais foram desenvolvidas, porém algumas tomaram um rumo mais parecido com as linguagens imperativas, adquirindo ferramentas que aumentam a eficiência de execução. LISP é um exemplo.

- ML
- Caml, SML, F#, Scheme

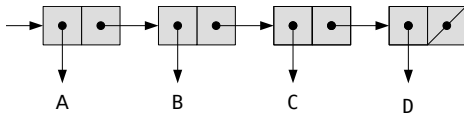
Tópicos

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

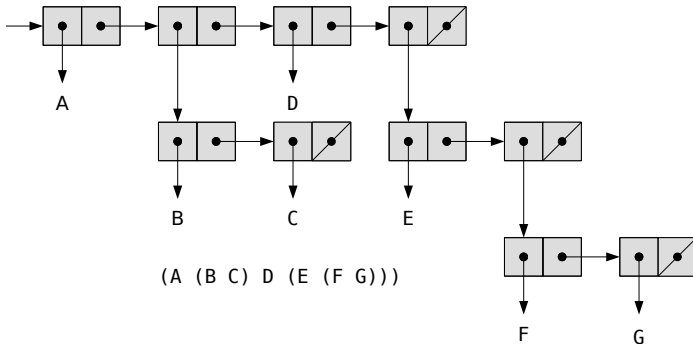
Tópicos

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

- Não existe tipagem de dados, porém existe dois tipos de objetos de dados: átomos e listas.
- Uma lista é composta por dois campos: o primeiro é o dado do elemento e o segundo pode ser um ponteiro para o próximo elemento ou uma lista vazia.



(A B C D)



(A (B C) D (E (F G)))

Funções do LISP também são expressas no formato de listas.

Exemplo

```
(defun double (x)
  (+ x x)
)
```

Tópicos

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

Existem dois tipos de funções: as primitivas e as definidas pelo usuário.

- Algumas primitivas sobre números:
+, -, /, *

Exemplo

```
(* 2 2)
```

Saída: 4

- Algumas funções sobre listas:
first, last, rest, append, list

Exemplo

```
(first '(A B C D))
```

- Aspa simples também é uma primitiva.

Tópicos

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

- Predicados são funções primitivas que retornam verdadeiro ou falso.
- Verdadeiro é denotado por T e falso por NIL.

Exemplo

```
(symbolp +)
```

Saída: T

- Outros predicados muito utilizados são os relacionais:
>, >=, <, <=, =, /=

Exemplo

(< 671 9)

Saída: NIL

Tópicos

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

LISP provê duas formas de controlar e organizar o fluxo de execução: condições e recursão.

Exemplo

```
(defun list-nth (n L)
  (cond
    ((null L) nil)
    ((zerop n) (first L))
    (t (list-nth (- n 1) (rest L))))
)
```

Tópicos

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

- É um dialeto de LISP onde foram implementadas ferramentas de linguagens imperativas:
if, case, while, for, etc

Exemplo

```
(defun fac (n)
  (if (= n 0) 1
      (* n (fac (- n 1)))
  )
)
```

Tópicos

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

- É uma linguagem de programação puramente funcional, ou seja, não possui expressões ou declarações que tem efeitos colaterais.

Exemplo

```
fact n
  | n == 0 = 1
  | n == 1 = 1
  | n > 1 = n * fact(n - 1)
```

Tópicos

- 1 Introdução
 - O Paradigma
 - Funções Matemáticas
 - Lambda Calculus
- 2 Noções Básicas
- 3 Linguagens
- 4 LISP
 - Listas
 - Funções Primitivas
 - Predicados
 - Estruturas de Controle
- 5 Common Lisp
- 6 Haskell
- 7 Conclusão

- Com base na não dependência de contexto, testes sobre funções são de fácil implementação.
- Porém, a aplicação pura do paradigma funcional é apenas para sistemas muito específicos.
- Além disso, para leigos em definições matemáticas haverá uma certa dificuldade inicial na modelagem de soluções para os problemas.



SEBESTA, Robert W. *Concepts of Programming Languages*. United States of America: Pearson, 10ed., 2010.