

# *Teoria de Funções Recursivas*

Professor Adolfo Neto  
DCC – UDESC  
Junho 2008

Obs.: Estes slides são um resumo do capítulo 1 do livro Modelos Clássicos de Computação, de Flávio S.C. da Silva e Ana C.V. De Melo.

---

---

# *Introdução/Motivação*

- Classe de funções de números naturais
  - Caracterizada por S. C. Kleene
  - Representa o que é possível resolver usando o processamento de um computador
  - Representa o conjunto de funções computáveis
  - Um problema pode ser resolvido usando o computador **se e somente se** puder ser traduzido na forma de uma função de números naturais pertencente a esta classe de funções
- 
-

# *Quem foi Kleene?*

- Stephen Cole Kleene
  - Matemático americano – 1909-1994
  - Aluno de Alonzo Church
  - Ph.D. em Matemática por Princeton
  - Professor da Universidade de Wisconsin-Madison
  - Fundador da Teoria da Recursão
  - Serviu na Marinha Americana durante a Segunda Guerra Mundial
  - Aposentou-se em 1979
  - Foi casado e teve 4 filhos. Escalava montanhas, praticava canoagem e foi um ambientalista.
- 
-

# *Números naturais*

- Caracterizados pelos postulados de Peano
  - Giuseppe Peano – matemático italiano -1858 – 1932
  - Propriedades como definição dos números naturais
- 
-

# Aritmética de Peano

- Propriedades como definição dos números naturais:
    - 0 é um número natural
    - Se  $x$  for natural, então existe um natural definido a partir de  $x$ :  $s(x)$  – o sucessor de  $x$
    - Para qq. n.nat.  $X$ ,  $0 \neq s(x)$
    - Se  $s(x)=s(y)$  então  $x=y$
    - Se uma propriedade  $Q$  for tal que:
      - 0 tem a propriedade  $Q$
      - Se  $x$  tem a propriedade  $Q$ , então  $s(x)$  também tem a propriedade  $Q$  para qq n.nat.  $X$ ;
    - então todos os n.nats.têm a propriedade  $Q$
- 
-

# Funções

- Denotaremos  $0, 1, 2, 3, \dots$  em vez de  $0, s(0), s(s(0)), s(s(s(0))), \dots$
  - Uma função  $n$ -ária de números naturais é uma função matemática de  $n$  argumentos.
  - Uma função  $n$ -ária de números naturais pode ser caracterizada por uma tabela com  $n+1$  colunas
  - Exemplos:
    - $f(x)=2*x$
    - $g(x,y)=x+y$
- 
-

# Algoritmos e Funções

- $h(x)$  = um número arbitrariamente selecionado para cada valor de  $x$ , sem respeitar qualquer regra de regularidade ou dependência dos valores específicos da variável  $x$ .
  - Para este exemplo não podemos construir um algoritmo que calcule os valores da função para quaisquer valores dos argumentos
  - Os valores da função que não forem explicitamente fornecidos permanecerão desconhecidos
  - Esta função não é computável.
- 
-

# *A proposta de Kleene*

- Um conjunto de funções iniciais e
  - Um conjunto de construtores de funções que podem ser encadeados
  - Uma função computável é uma função inicial ou uma função obtida a partir das funções iniciais por meio do encadeamento dos construtores.
- 
-

# Funções recursivas primitivas (FRPs)

- Funções iniciais:
  - $Ze(x)=0$
  - $Suc(x)=s(x)$
  - $Pr^n_i(x_1, \dots, x_n)=x_i$
- O conjunto das f.r.p.s é o MENOR conjunto composto por:
  - Funções iniciais
  - Função resultante de um número finito de aplicações dos seguintes construtores sobre frps:
    - Substituição
    - Recursão

# Funções recursivas primitivas (FRPs)

- Substituição
  - Se  $g(x_1, \dots, x_m)$ ,  $h_1(x_1, \dots, x_n)$ , ...,  $h_m(x_1, \dots, x_n)$  são funções
  - Então construímos
    - $f(x_1, \dots, x_n) = g(h_1(x_1, \dots, x_n), \dots, h_m(x_1, \dots, x_n))$
- Recursão
  - Se  $g(x_1, \dots, x_n)$  e  $h(x_1, \dots, x_n, y_1, y_2)$  são funções
  - Então construímos:
    - $f(x_1, \dots, x_n, 0) = g(x_1, \dots, x_n)$
    - $f(x_1, \dots, x_n, s(y)) = h(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y))$ 
      - Observe que  $n$  pode valer 0

# *Exemplos de FRPs*

- Função constante
- Função soma
- $f(x)=k*x$



# Considerações

- É bastante trabalhoso verificar se uma função é recursiva primitiva
- A dificuldade cresce ainda mais quando o resultado esperado é negativo

# *Funções computáveis que não são recursivas primitivas*

- Existem funções que embora sejam computáveis (ou seja, sabemos construir um programa para calcular o seu valor para quaisquer números naturais) não são recursivas primitivas
- Ex.: função de Ackermann



# Funções recursivas totais

- São as FRPs estendidas com o construtor  $\mu$ :
    - Seja  $g(x_1, \dots, x_n, y)$  uma função tal que para quaisquer valores de  $x_1, \dots, x_n$  existe pelo menos um valor de  $y$  de modo que  $g(x_1, \dots, x_n, y) = 0$ .
    - Seja  $\mu_y [g(x_1, \dots, x_n, y) = 0]$  o menor valor de  $y$  para que  $g(x_1, \dots, x_n, y) = 0$ .
    - Então, apenas para as funções  $g(x_1, \dots, x_n, y)$  com esta propriedade, podemos construir a função  $f(x_1, \dots, x_n) = \mu_y [g(x_1, \dots, x_n, y) = 0]$ .
- 
-

# *Funções parciais*

- Programação é perfeitamente factível
- Também deveriam ser consideradas computáveis



# Funções recursivas parciais

- São as FRPs estendidas com o construtor  $\mu$ :
  - Seja  $g(x_1, \dots, x_n, y)$  uma função de números inteiros.
  - Seja  $\mu_y [g(x_1, \dots, x_n, y)=0]$  o menor valor de  $y$  – se existir – tal que  $g(x_1, \dots, x_n, y)=0$ .
  - Então, podemos construir a função  $f(x_1, \dots, x_n) = \mu_y [g(x_1, \dots, x_n, y)=0]$ .

# *Funções recursivas parciais*

- Generalização do construtor de minimização
- Correspondem exatamente ao conjunto de funções computáveis
- O mesmo conjunto definido por:
  - Máquinas de Turing
  - Máquinas de Post
  - Lambda-Cálculo
  - etc.

# ***Bibliografia***

- **SILVA, F.S.C. Da & MELO, A.C.V de. Modelos Clássicos de Computação.** São Paulo: Thomson Learning, 2006.
- **MENDELSON, E. Introduction to mathematical logic.** Fourth Edition. CRC Press, 1997.
- **WIKIPEDIA. Stephen Cole Kleene.** Disponível em [http://en.wikipedia.org/wiki/Stephen\\_Cole\\_Kleene](http://en.wikipedia.org/wiki/Stephen_Cole_Kleene). Acessado em 18 de junho de 2008.

# *Bibliografia*

- CARNIELLI, W.A. & EPSTEIN, R.L.  
**Computabilidade, funções computáveis, lógica e os fundamentos da matemática.** São Paulo: Editora UNESP, 2006.