



# TEC0001 – Teoria da Computação

## Aula 01

### Computabilidade

Karina Girardi Roggia  
karina.roggia@udesc.br

Departamento de Ciência da Computação  
Centro de Ciências Tecnológicas  
Universidade do Estado de Santa Catarina

2019

# Sumário

Introdução

Algoritmo

Modelos





# Perguntas...

O que é um algoritmo?

O que é *computar*?

Quais os limites do que é computável?





# Algoritmo

*Um algoritmo é um conjunto de instruções seguidas por um computador*



# Algoritmo

*Um procedimento para resolver um problema matemático em um número finito de passos que frequentemente envolve repetições de operações. De forma ampla, um método passo a passo para cumprir uma tarefa.*



# Algoritmo

Uma coleção bem ordenada de operações não ambíguas e efetivamente computáveis que, quando executadas, produzem um resultado e param em um período de tempo finito.



# Algoritmo

Primeiros algoritmos:

- Euclides (325 – 265 A.C.)
- Al Khwarizmi (780 – 850)

Computabilidade Efetiva

- Church – Cálculo Lambda – Maio/1935
- Gödel – Funções Recursivas – Julho/1935
  - Formalizado por Kleene em 1936
- Turing – Máquinas de Turing – Maio/1936

# Os Problemas de Hilbert

## II Congresso Internacional de Matemáticos (1900, Paris)

- Hilbert apresentou 10 problemas em aberto, de uma lista de 23 que ele divulgou posteriormente, que ele julgava cruciais para o desenvolvimento da matemática moderna.
- 2º Problema: Demonstrar a consistência dos axiomas da aritmética.
- A solução deste problema conflitou com um dos grandes propósitos de Hilbert
  - Definição da matemática através da lógica, usando o método de sistemas formais, ou seja, provas finitistas a partir de um conjunto constituído por axiomas.





# Entscheidungsproblem

Obter um algoritmo que, dada uma afirmação em lógica formal, determine se tal afirmação é verdadeira ou falsa.

Ideia de que a lógica é **completa**: Toda afirmação que é verdadeira pode ser demonstrada.

Esta afirmação não é demonstrável.

# Cálculo Lambda

$$\begin{array}{lcl} L, M, N & ::= & x \\ & | & (\lambda x. N) \\ & | & (L M) \end{array}$$

# Funções Recursivas Parciais

Função Constante

$$f(x_1, \dots, x_k) = n$$

Função Sucessor

$$S(x) \stackrel{\text{def}}{=} f(x) = x + 1$$

Função de Projeção

$$P_i^k \stackrel{\text{def}}{=} f(x_1, \dots, x_k) = x_i$$

Operador de Composição

$$h \circ (g_1, \dots, g_m) \stackrel{\text{def}}{=} f(x_1, \dots, x_k) = h(g_1(x_1, \dots, x_k), \dots, g_m(x_1, \dots, x_k))$$

Operador Primitivo Recursivo

$$\rho(g, h) \stackrel{\text{def}}{=} f(y, x_1, \dots, x_k) \text{ onde}$$

$$f(0, x_1, \dots, x_k) = g(x_1, \dots, x_k)$$

$$f(z + 1, x_1, \dots, x_k) = h(z, f(z, x_1, \dots, x_k), x_1, \dots, x_k)$$

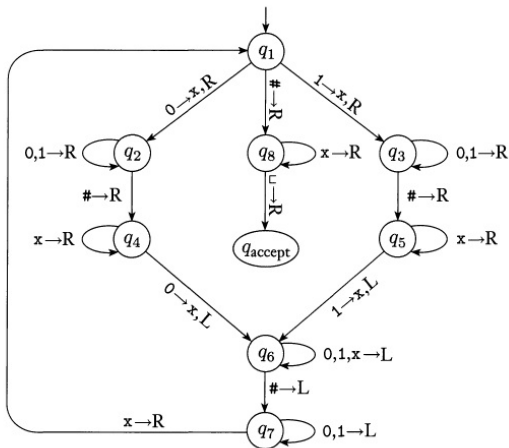
Operador de Minimização

$$\mu(f)(x_1, \dots, x_k) = z \stackrel{\text{def}}{\Leftrightarrow} f(z, x_1, \dots, x_k) = 0 \text{ e}$$

$$f(i, x_1, \dots, x_k) > 0 \text{ para } i = 0, \dots, z - 1$$

# Máquinas de Turing

$$M = \langle Q, \Sigma, \Gamma, \delta, q_1, q_{\text{accept}}, q_{\text{reject}} \rangle$$





# Pipoca!

Philip Wadler - Propositions as types

<https://www.youtube.com/watch?v=IOiZatIZtGU>

(Parte I, até o instante 08:55)

