

## Lógica da Computação - 2020/1

Aula 17/T08 - 04/May/2020

Raul Ikeda - rauligs@insper.edu.br

### Objetivos

1. Computabilidade e Decidibilidade
2. Problema da Parada

**Exercício 1:** Fazer uma Máquina de Turing para somar dois números.

**Exercício 2:** Fazer uma Máquina de Turing para multiplicar dois números.

**Exercício 3:** Fazer uma Máquina de Turing para somar dois números binários.

**Afirmção:** “Todo computador pode simular uma Máquina de Turing e toda Máquina de Turing pode simular um computador.”

>> Ver Hopcroft et. al. Pag. 381.

### Entscheidungsproblem

Linha do Tempo:

- 400 a.C. - Formas primitivas de ábaco na China e na Babilônia.
- 100 a.C. - Máquina de Anticítera.
- 1623 - Schikard: primeiro relógio de calcular.
- 1630 - Oughtred: régua de cálculo.
- 1642 - Pascal: máquina de calcular de 8 dígitos.
- 1673 - Leibniz: máquina de calcular mais simples e eficiente (incluindo raiz quadrada).
- 1800 - Jacquard: cartão perfurado.
- 1823 - Babbage: máquina de diferenças nº 1.
- 1854 - Boole: lógica binária.
- 1896 - Hollerith: máquina de leitura de cartões para censo.
- 1936 - Turing: “On computable numbers, with an application to the Entscheidungsproblem”.

## Origens da Computabilidade:

Em 1900 David Hilbert, famoso matemático alemão, elencou 23 problemas matemáticos para o século XX. Em especial, o problema número 10 dizia respeito a um “algoritmo” para dizer um polinômio possuía uma raiz inteira.

A frase exata dito por Hilbert foi: “Given a Diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: To devise a process according to which it can be determined in a finite number of operations whether the equation is solvable in rational integers”. Podemos entender por *process* e *finite number of operations* como a primeira definição de **algoritmo**.

**Último Teorema de Fermat:** Dados  $x, y$  e  $z$  números inteiros, não existe um número inteiro  $n$  maior que 2 tal que  $x^n + y^n = z^n$ .

**Exercício 4:** Escreva um programa em linguagem de alto nível para verificar o último teorema de Fermat.

Na década de 1930, Hilbert lançou *Hilbert's Entscheidungsproblem* (problema de decisão de Hilbert), onde ele acreditava que não existiria um problema insolúvel, era só questão de achar o algoritmo correto.

Para analisar o problema, era preciso primeiro definir o que é um algoritmo. Alonzo Church (1936) e Allan Turing (1936) publicaram separadamente a noção de algoritmo através das suas “máquinas”: cálculo- $\lambda$  e máquina de Turing respectivamente. Ambas as definições demonstraram-se equivalentes e ficaram conhecidas como a **Tese de Church-Turing**.

Martin Davis, Yuri Matiyasevich, Hilary Putnam e Julia Robinson provaram em 1970, em um trabalho que durou 21 anos, que a resposta do décimo problema era negativa. Ou seja, não há um algoritmo e portanto o problema é **indecidível**.

## A Tese de Church-Turing e os Problemas Decidíveis

Teorema: “Um AFD que reconhece ou não uma certa cadeia é um problema decidível”.

**Exercícios:** Estender o conceito acima para as seguintes questões decidíveis:

1.  $w \in AFN$ ?
2. Uma expressão regular gera uma certa cadeia  $w$ ?
3.  $L(AFD) = \emptyset$ ?
4.  $L(AFD_1) = L(AFD_2)$ ?
5.  $w \in GLC$ ?
6.  $L(GLC) = \emptyset$ ?
7.  $L(GLC_1) = L(GLC_2)$ ?
8. Toda LLC é decidível?

### O Problema da Parada

Teorema: “Uma MT que aceita ou não uma cadeia é indecidível”.

>> Ver Sipser Pag. 182. Hopcroft et. al. Pag. 328.

**Exercício:** É indecidível se uma GLC é ou não ambigua.

Corolário: “ $\bar{L}_p$  não é Recursivamente Enumerável.”

>> Ver Sipser Pag. 191.

**Próxima aula:**

Continuação do roteiro H08.