Introdução à Teoria da Computação

Teoria da Computação

INF05501

Ciência da Computação

- Ciência da Computação é o conhecimento **sistematizado** relativo à **computação**
- Origens:
  - Na antiga Grécia (século III A.C.), no estudo de **algoritmos** por Euclides
  - Na Babilônia, com estudos sobre a **complexidade** e a **reducibilidade** de **problemas**

Ênfases

- Interesse atual possui **duas ênfases**:

  1. Teórica: ideias fundamentais e **modelos computacionais**
     - Modelos para redes de neurônios (Biologia)
     - Teoria do chaveamento (Eletrônica)
     - Lógica (Matemática)
     - Gramáticas para linguagens naturais (Linguística)
     - ...
  2. Prática: projeto de sistemas computacionais, aplicando as **ideias da teoria na prática**

Teoria da Computação

- Ramo da Ciência da Computação, que dado um **problema**, determina
  - Se tal problema pode ser resolvido computacionalmente → Teoria da Computabilidade
  - Com quê eficiência ele pode ser resolvido → Teoria da Complexidade
- Esta **solução computacional** envolve
  - Um modelo computacional
  - Um algoritmo

Modelos Computacionais

- No início do século XX, diversas pesquisas foram desenvolvidas com o objetivo de definir um **modelo computacional** suficientemente **genérico**

- Tal modelo computacional deveria ser capaz de implementar **qualquer função computável**

Modelos Computacionais (cont.)

- Nesta busca, alguns **modelos de aplicação mais restrita** foram propostos, tais como:
  - Expressões regulares (padrões de cadeias de caracteres)
  - Gramáticas Livres de Contexto (sintaxe de linguagens de programação)
  - Autômatos Finitos (projeto de hardware e software)

Modelos Computacionais (cont.)

- Para **medir o poder computacional** de um modelo, temos de identificar a **linguagem formal que ele pode gerar**

- Desta forma, a **comparação entre diferentes modelos** permite estabelecer-se uma hierarquia de modelos

Notas Históricas

- **Marco inicial:** David Hilbert com seu trabalho ***Entscheidungsproblem*** *(problema de decisão)* em 1928

- Perguntava se existia um **algoritmo** capaz de receber uma **descrição de uma linguagem formal** qualquer e uma **proposição matemática** qualquer nesta linguagem e **decidir se a proposição é falsa ou verdadeira**

- Tal problema foi particularizado para a questão de se encontrar um **procedimento** para demonstrar se uma dada **fórmula no cálculo de predicados era válida ou não**

Notas Históricas (cont.)

- Em 1931, Kurt Gödel apresentou os Teoremas da Incompletude
- Determinava que a **mecanização** do processo de **prova de teoremas** não tem solução
- Neste trabalho, Gödel utilizou números naturais para **codificar** símbolos e fórmulas (“aritmeticanização da sintaxe”)
- Na prova dos teoremas, afirmações sobre fórmulas foram descritas como **funções primitivas recursivas** (definidas por Dedekind em 1888)
- Gödel foi (aparentemente) o primeiro a identificar um **formalismo para definir a noção de procedimento efetivo**

Notas Históricas (cont.)

- Em 1936, Alonzo Church usou dois formalismos para mostrar que o **problema de Hilbert não tem solução**:
  - Cálculo λ (Church, 1936)
  - Funções recursivas (Kleene, 1936)
- No mesmo ano, Stephen Kleene verificou a **equivalência destes formalismos**

Procedimentos Efetivos

"Tais formalismos são caracterizações tão gerais da noção do efetivamente computável quanto consistentes com o entendimento intuitivo usual" (Hipótese de Church)

- Esta hipótese é **tida como verdadeira** e usada como base de muitos trabalhos na área de Teoria da Computação

- No entanto, ela não pode ser provada, pois parte de uma **noção intuitiva**

Procedimentos Efetivos (cont.)

- Alan Turing propôs, em 1936, um **formalismo para a representação de procedimentos efetivos**
- Ideia era **simular**, computacionalmente, **atitudes humanas ao realizar cálculos**
- Foi o primeiro trabalho a identificar programas escritos para uma "máquina computacional" como **noções intuitivas de efetividade**

Procedimentos Efetivos (cont.)

- Desde então, muitos outros formalismos com o **mesmo poder computacional que as funções recursivas** (ou que o Cálculo λ) foram propostos
  - Máquina de Turing (1936)
  - Sistema Canônico de Post (1943)
  - Algoritmo de Markov e a Linguagem Snobol (1954)
  - Máquinas de Registradores (1963)
  - RASP (*Random Acess Stored Programs* - 1964)

Procedimentos Efetivos (cont.)

- **Programa** é definido como sendo um **procedimento efetivo**
- Pode ser descrito usando qualquer dos **formalismos equivalentes**
- Logo, qualquer destes formalismos permite descrever **todos os procedimentos possíveis de serem executados em um computador**
- Com isto, pode-se conhecer os limites do quê é computável