

Introdução à Teoria da Computação

Aula 01

Professor Luís Carlos Pompeu

Conteúdo da aula

- Introdução à Teoria da Computação
- Conceitos iniciais sobre linguagens formais, máquinas e automação.
- Gramáticas Formais
- Definição de gramáticas, tipos de gramáticas (Chomsky), produção de linguagens formais.

Apresentação

- Sobre mim:
 - Formação;
 - Atuação profissional;

Apresentação

- Sobre a disciplina:
- Por que estudar Teoria da Computação?
 - Teoria da computação, linguagens formais e autômatos.
- Teoria das linguagens formais:
 - Sintaxe e semântica;
 - Abordagem;
 - Formalismo operacional;
 - Formalismo axiomático;
 - Formalismo denotacional.

Por que estudar Teoria da Computação?

- Contemplam tópicos que os alunos (geralmente) consideram:
- Excessivamente áridos (principalmente por sua proximidade com a matemática);
- Abstratos;
- Complexos;
- Desvinculados de sua futura realidade profissional.

Por que estudar Teoria da Computação?

- Se por um lado...
 - Cursos que possuem maior ênfase nos aspectos tecnológicos da computação.
 - Favorecem a empregabilidade do aluno recém formado.
- Por outro lado...
 - Esses alunos se ressentem, à medida que o tempo passa, da falta de uma formação teórica mais completa.
 - Que lhes permita se renovarem e se manterem competitivos no mercado de trabalho por mais tempo.

Por que estudar Teoria da Computação?

- Teoria da Computação
 - Compreende as propriedades matemáticas fundamentais do hardware, software, e das aplicações de computadores.
 - Mostra um lado mais simples, e mais elegante computadores.
- Teoria é relevante para a prática.
 - Provê ferramentas conceituais uteis.
 - Projetar uma nova linguagem de programação para uma aplicação especializada:
 - Gramáticas.
- Lidar com buscas por strings e casamento de padrões:
 - Autômatos finitos e Expressões regulares.

Por que estudar Teoria da Computação?

- Teoria é bom porque expande a sua mente.
- A tecnologia de computadores muda rapidamente.
 - Conhecimento técnico específico, embora útil hoje, fica desatualizado em poucos anos.
- Habilidade de pensar, exprimir-se claramente e precisamente para resolver problemas (principalmente matemáticos), e saber quando você não resolveu um problema.
 - Essas habilidades possuem valor duradouro.



Por que estudar Teoria da Computação?

- Teoria é bom porque expande a sua mente.
- A tecnologia de computadores muda rapidamente.
 - Conhecimento técnico específico, embora útil hoje, fica desatualizado em poucos anos.
- Habilidade de pensar, exprimir-se claramente e precisamente para resolver problemas (principalmente matemáticos), e saber quando você não resolveu um problema.
 - Essas habilidades possuem valor duradouro.

Por que estudar Teoria da Computação?

- **Teoria dos autômatos** (é o estudo das máquinas abstratas, bem como problemas computacionais que podem ser resolvidos usando esses objetos. É objeto de estudo tanto da Ciência da Computação Teórica como da Matemática Discreta.)
 - Permite criar definições e propriedades de modelos matemáticos de computação.
- **Autômato Finito** (modelos mais simples, saem de um estado finito para outro estado finito)
 - Usado em processamento de texto, compiladores, e projeto de hardware
- **Gramática Livre de Contexto**
 - Usado em linguagens de programação e inteligência artificial.

Por que estudar Teoria da Computação?

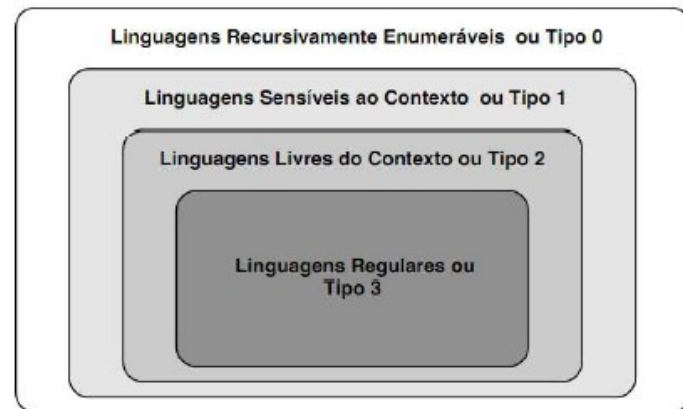
- Máquina de Turing
 - Modelo matemático preciso de um computador, porém simplificado.
 - Permite estudar pontos importantes da teoria da computação, como: Teoria da Computabilidade e da Complexidade
 - Em 1930, antes de existirem computadores, Alan Turing desenvolveu uma máquina abstrata que tinha todas as características dos computadores atuais, ao menos no que se refere ao quanto eles poderiam calcular. O objetivo de Turing era descrever com exatidão o que uma máquina de computação seria e o que ela não seria capaz de fazer. As conclusões de Turing se aplicam não apenas a sua máquina, mas também as máquinas reais de hoje.
 - **Computabilidade**: Podemos definir um problema como "**computável**" se ele puder ser resolvido pela máquina de Turing.
 - **Complexidade**: Determinado que um problema é computável, qual o custo computacional para resolução.



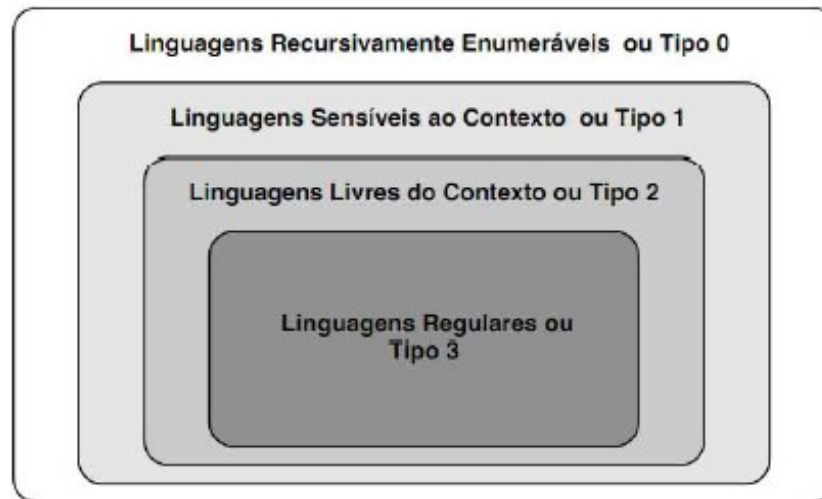
Teoria da computação: Linguagens formais

- Teoria das Linguagens Formais (São linguagens que podem ser representadas de maneira finita e precisa através de sistemas com sustentação matemática)
 - Trata da caracterização, classificação e formalização, e propriedades das linguagens estruturadas em frases.
- Hierarquia de Chomsky
 - Classificação das linguagens estruturadas em frases, organizadas em níveis de complexidade crescente.

A complexidade cresce de dentro pra fora.
Qualquer problema não computável estará fora dessa hierarquia



Teoria da computação: Linguagens formais



- Hierarquia de Chomsky:
 - Tal organização proporciona ao aluno uma introdução natural e gradual aos assuntos da área.
 - À medida que o estudo das classes de linguagens mais simples dá lugar ao estudo das classes de linguagens mais complexas.

Teoria da computação: Linguagens formais

- Linguagens Formais:
 - Oferece uma perspectiva de estudo baseada na **síntese** de cadeias
 - Dispositivos gramaticais.
- Teoria dos Autômatos:
 - Oferece uma perspectiva de estudo baseada na **análise** de cadeias
 - Dispositivos de reconhecimento.

Teoria das Linguagens formais

- Desenvolvida na década de 1950
 - Objetivo inicial:
 - Desenvolver teorias relacionadas com as **linguagens naturais**. (estudo de linguística, ou seja, como se formam as línguas ou idiomas)
 - Entretanto, logo foi verificado que era importante para o estudo de **linguagens artificiais**.
 - Em especial, para as linguagens originárias da Computação e Informática.
 - Desde então, desenvolveu-se significativamente...

Teoria das Linguagens formais

- Desenvolvida na década de 1950
 - Objetivo inicial:
 - Desenvolver teorias relacionadas com as **linguagens naturais**.
(estudo de linguística, ou seja, como se formam as línguas ou idiomas)
 - Entretanto, logo foi verificado que era importante para o estudo de **linguagens artificiais**. (inicialmente os programadores eram obrigados a escrever programas na linguagem do computador)
 - Em especial, para as linguagens originárias da Computação e Informática.
 - Desde então, desenvolveu-se significativamente...
 - Das linguagens de baixo nível até as linguagens de alto nível

Teoria das Linguagens formais

- Exemplos de aplicações
 - **Análise léxica e análise sintática** de linguagens de programação; léxica: analisa, por exemplo: se o conteúdo de uma variável do tipo int contém um inteiro. Sintática: analisa se um comando foi escrito corretamente
 - Análise léxica e análise sintática em processadores de texto;
 - Modelagem de circuitos lógicos ou redes lógicas;
 - Modelagem de sistemas biológicos:
 - Autômatos celulares.
- Mais recentemente:
 - Animações;
 - Hipertextos e hipermídias:
 - Criptografia:
 - Etc.

Teoria das Linguagens formais

- Linguagens Formais preocupa-se com os problemas sintáticos das linguagens
 - Conceitos de Sintaxe e **Semântica** (seu código pode estar sintaticamente correto e não resolver o seu problema)
- Historicamente, o problema sintático:
 - Reconhecido antes do problema semântico;
 - Primeiro a receber um tratamento adequado;
 - Tratamento mais simples que os semânticos.

Teoria das Linguagens formais

- Consequência:
 - Grande ênfase à sintaxe;
 - Levando à ideia de que questões das linguagens de programação
 - Resumiam-se às questões da sintaxe.
- Teoria da sintaxe
 - Possui construções matemáticas bem definidas e universalmente reconhecidas
 - [EX] Gramáticas de Chomsky

Sintaxe e Semântica

- Linguagem de programação (ou qq modelo matemático) pode ser vista como:
 - Uma entidade livre, sem qualquer significado associado.
 - ou
 - Juntamente com uma interpretação do seu significado.
- Sintaxe:
 - Trata das propriedades livres da linguagem.
 - [EX] verificação gramatical de programas.
- Semântica:
 - Objetiva dar uma interpretação para a linguagem.
 - [EX] significado ou valor para um determinado programa.

Sintaxe e Semântica

- Linguagem de programação (ou qq modelo matemático) pode ser vista como:
 - Uma entidade livre, sem qualquer significado associado.
 - ou
 - Juntamente com uma interpretação do seu significado.
- Sintaxe:
 - Trata das propriedades livres da linguagem.
 - [EX] verificação gramatical de programas.
- Semântica:
 - Objetiva dar uma interpretação para a linguagem.
 - [EX] significado ou valor para um determinado programa.

Sintaxe e Semântica

- Consequentemente, a sintaxe:
 - manipula símbolos, sem considerar os seus correspondentes significados
- Mas, para resolver qualquer problema real:
 - Necessário dar uma interpretação semântica aos símbolos
 - [EX] estes símbolos representam os inteiros: 1, “2”, “3”...
- Não existe a noção de programa sintaticamente "errado"
 - Simplesmente não é um programa daquela linguagem.
- Sintaticamente válido ("correto"):
 - Pode não ser o programa que o programador esperava escrever.

Sintaxe e Semântica

- Programa "correto" ou "errado"
 - Se o mesmo modela adequadamente o comportamento desejado
- Limites entre a sintaxe e a semântica:
 - Nem sempre são claros em linguagens naturais.
 - Entretanto, a distinção entre sintaxe e semântica é, em geral, óbvia em linguagens artificiais.
- Análise léxica
 - Tipo especial de análise sintática;
 - Centrada nas componentes básicas da linguagem;
 - Portanto, também é ênfase das Linguagens Formais.

Sintaxe e Semântica

- Autômatos finitos, passa de um estado finito a outro estado finito: também chamado de arestas

