

[Aula 01] Introdução à teoria da computação – Conceitos básicos

Prof. João F. Mari
joaof.mari@ufv.br

BIBLIOGRAFIA

- SIPSER, M. **Introdução a Teoria da Computação**, 1. Ed., Thomson Pioneira, 2007
- MENEZES, P.B. **Linguagens formais e autômatos**, 6. ed., Bookman, 2011.
 - Capítulo 1
- RAMOS, M.V.M.R. Ensino de linguagens formais e autômatos em cursos de computação. **Revista de Computação e Tecnologia da PUC-SP**, 2009.
 - Disponível em:
 - <http://revistas.pucsp.br/index.php/ReCET/article/view/1033>



ROTEIRO

- 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?

- Teoria da Computação, Teoria de Linguagens Formais e Teoria dos Autômatos.
- Contemplam tópicos que os alunos (geralmente) consideram:
 - Excessivamente áridos;
 - 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 dos 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?

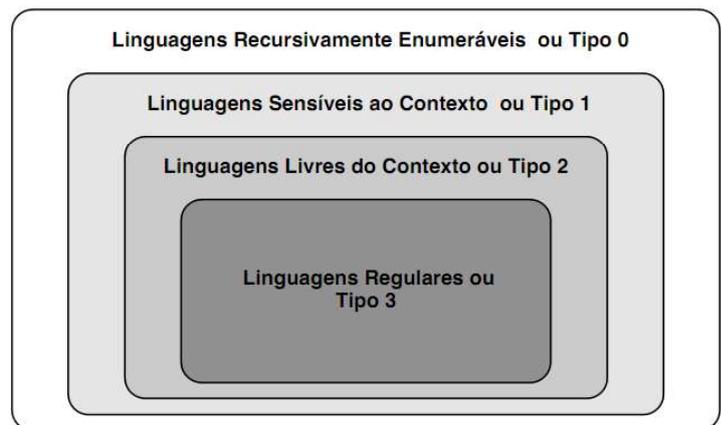
- Teoria é bom porque expande a sua mente.
- A tecnologia de computadores muda rapidamente.
 - Conhecimento técnico específico, embora útil hoje, fica desatualizado em poucos anos.
- Habilidade de pensar, exprimir-se claramente e precisamente para resolver problemas, e saber quando você não resolveu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

Por que estudar Teoria da Computação?

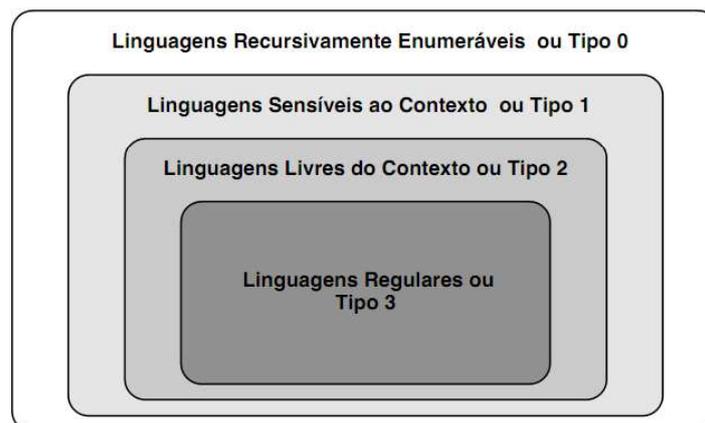
- Teoria dos autômatos
 - Definições e propriedades de modelos matemáticos de computação.
- Autômato Finito
 - Usado em processamento de texto, compiladores, e projeto de hardware
- Gramática Livre de Contexto
 - Usado em linguagens de programação e inteligência artificial.
- Máquina de Turing
 - Modelo matemático preciso de um computador, porém simplificado.
 - Permite estudar pontos importantes da teoria da computação, como: Teoria da Computabilidade e da Complexidade

Teoria da computação, linguagens formais e autômatos

- Teoria das Linguagens Formais
 - 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.



Teoria da computação, linguagens formais e autômatos



- 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 e autômatos

- **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 da computação, linguagens formais e autômatos

	ANÁLISE	SÍNTESE
Linguagens regulares	Autômatos finitos	Gramática regular
Linguagens livres de contexto	Autômatos com pilha	Gramática livre de contexto
Linguagens sensíveis ao contexto	Máquina de Turing com fita limitada	Gramática sensível ao contexto
Linguagens recursivamente enumeráveis	Máquina de Turing	Gramática irrestrita
Linguagens recursivas	Máquina de Turing que sempre para	...

Teoria das Linguagens Formais

Teoria das Linguagens Formais

- Desenvolvida na década de 1950
 - Objetivo inicial:
 - Desenvolver teorias relacionadas com as **linguagens naturais**.
 - 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

- Exemplos de aplicações
 - **Análise léxica** e **análise sintática** de linguagens de programação;
 - **Análise léxica** e **análise sintática** em processadores de texto;
 - Modelagem de circuitos lógicos ou redes lógicas;
 - Modelagem de sistemas biológicos:
 - Autômatos celulares:
 - https://en.wikipedia.org/wiki/Cellular_automaton
 - Jogo da vida:
 - https://en.wikipedia.org/wiki/Conway's_Game_of_Life
 - Etc.
- Mais recentemente:
 - Animações;
 - Hipertextos e hiperlinks;
 - Criptografia;
 - Etc.

Sintaxe e semântica

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

- Conseqüentemente, 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', '4', ...
- 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.

Abordagem

- Centrada no tratamento **sintático**
 - Linguagens lineares abstratas;
 - Com fácil associação às linguagens da Computação e Informática.
- Classificação dos **formalismos**:
 - Operacional;
 - Axiomático;
 - Denotacional.

Formalismo operacional

- Autômato ou uma máquina abstrata:
 - Estados e Instruções primitivas;
 - Especificação de como cada instrução modifica cada estado.
- Máquina abstrata
 - Suficientemente simples para não permitir dúvidas sobre a execução de seu código.
- Também é dito um formalismo **reconhecedor**:
 - Análise de uma entrada para verificar se é "reconhecida".
- Principais autômatos:
 - Autômato Finito;
 - Autômato com Pilha;
 - Máquina de Turing.

Formalismo axiomático

- Associam-se regras às componentes da linguagem.
- Regras permitem afirmar:
 - O que será verdadeiro após a ocorrência de cada cláusula, considerando-se o que era verdadeiro antes da ocorrência.
- Também é dito um formalismo **gerador**:
 - Verifica se um elemento da linguagem é "gerado"
- Gramáticas:
 - Regulares;
 - Livres do Contexto;
 - Sensíveis ao Contexto;
 - Irrestritas.

Formalismo denotacional (ou funcional)

- Define-se um domínio:
 - Caracteriza o conjunto de palavras admissíveis na linguagem;
 - Funções, em geral, composicionais (horizontalmente):
 - Valor denotado por uma construção
 - Especificado em termos dos valores denotados por suas subcomponentes.
- Abordagem restrita às **Expressões Regulares**.
- Também é dito um formalismo **gerador**:
 - É simples inferir ("gerar") as palavras da linguagem.

[FIM]

- FIM:
 - **[AULA 01]** Introdução à teoria da computação e conceito básicos
- Próxima aula:
 - **[AULA 02]** Linguagens e Gramáticas