

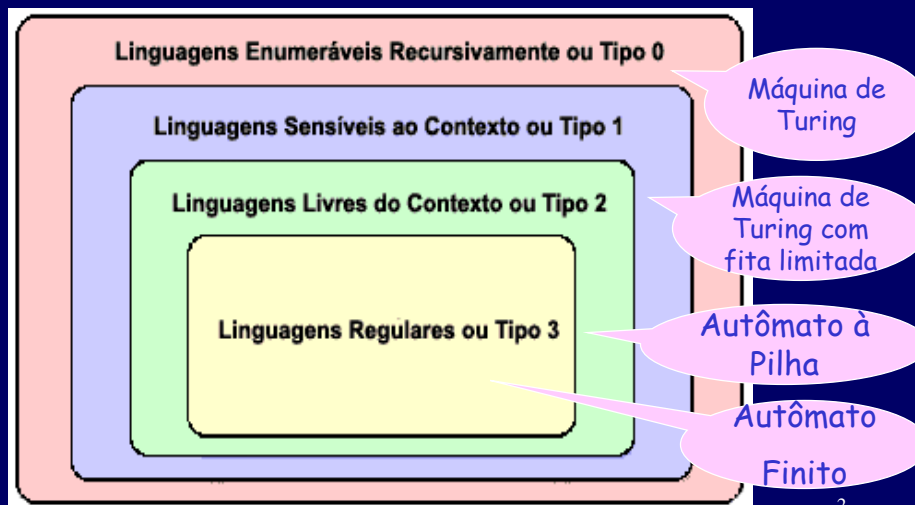
Linguagens Formais e Autômatos

Hierarquia de Chomsky:
Linguagens Regulares, Livres de Contexto, Sensíveis ao Contexto e Irrestritas
seus Reconhecedores, e Geradores

Autômatos são essenciais para o estudo dos limites da computação.
Existem dois limites estudados aqui:
O que um computador pode fazer?
O que um computador pode fazer eficientemente?

1

Hierarquia de Chomsky e sua relação com os autômatos



LFA e a Teoria da Computabilidade

Todas as linguagens recursivas são RE.

Todas as regulares, livres de contexto e sensíveis ao contexto são recursivas

Máquina de Turing que reconhece

| Automata theory: formal languages and formal grammars | | | |
|---|-------------------|------------------------|-------------------|
| Chomsky hierarchy | Grammars | Languages | Minimal automaton |
| Type-0 | (unrestricted) | Recursively enumerable | Turing machine |
| | (unrestricted) | Recursive | Decider |
| Type-1 | Context-sensitive | Context-sensitive | Linear-bounded |
| Type-2 | Context-free | Context-free | Pushdown |
| Type-3 | Regular | Regular | Finite |

Each category of languages or grammars is a proper superset of the category directly beneath it.

Máquina de Turing que Decide

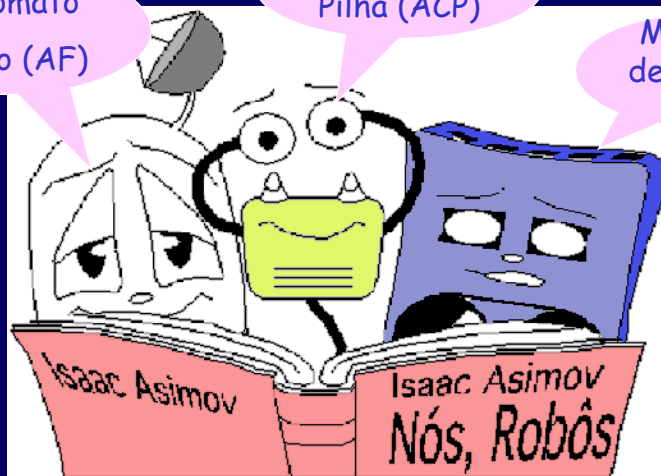
3

Autômatos Estudados

Autômato Finito (AF)

Autômato à Pilha (ACP)

Máquina de Turing (MT)



4


Exemplos de AF's

- Pensem em:
 - um controlador automático de portas de shoppings
 - um controlador de elevador
 - um interruptor
- AF tem um número FINITO de estados , cujo propósito é lembrar de porções relevantes da história do sistema.
 - A história inteira não pode ser lembrada ...uma decisão importante de projeto é o que os estados significam.

5

JFLAP
<http://www.jflap.org>
<http://pt.wikihow.com/Baixar-e-Usar-o-JFLAP>

JFLAP
Version 6.4
RELEASED
July 13, 2008



Não veremos

Autômatos

Gramáticas

Expressões Regulares - denotam linguagens regulares

6

Detailed description: The image shows a screenshot of the JFLAP software's 'New Document' dialog box. The dialog has a menu bar with 'File', 'Help', 'Batch', and 'Preferences'. Below the menu bar is a list of options: 'Finite Automaton', 'Mealy Machine', 'Moore Machine', 'Pushdown Automaton', 'Turing Machine', 'Multi-Tape Turing Machine', 'Grammar', 'L-System', 'Regular Expression', 'Regular Pumping Lemma', and 'Context-Free Pumping Lemma'. On the left, a bracket labeled 'Não veremos' (We won't see) points to the first three options. On the right, a bracket labeled 'Autômatos' (Automata) points to the first five options. Another bracket labeled 'Gramáticas' (Grammars) points to 'Grammar' and 'L-System'. A third bracket labeled 'Expressões Regulares - denotam linguagens regulares' (Regular Expressions - denote regular languages) points to 'Regular Expression'. The number '6' is at the bottom right of the screenshot area.

Usando JFLAP com AF: interruptor "onofre"

The screenshot shows the JFLAP interface. At the top, a finite automaton diagram has two states: q_0 (labeled "off") and q_1 (labeled "on"). Transitions are labeled "ligar" (from q_0 to q_1) and "desligar" (from q_1 to q_0). Below the diagram is a window titled "Accepting configuration found!" containing a list of configurations: q_0 "ligar" "desligar" "ligar", q_1 "ligar" "desligar" "ligar", q_0 "ligar" "desligar" "ligar", and q_1 "ligar" "desligar" "ligar". A pink callout box points to this window with the text "representam as informações externas sobre o sistema". Another pink callout box points to the transitions with the text "Entrada: 'ligar' 'desligar' 'ligar'".

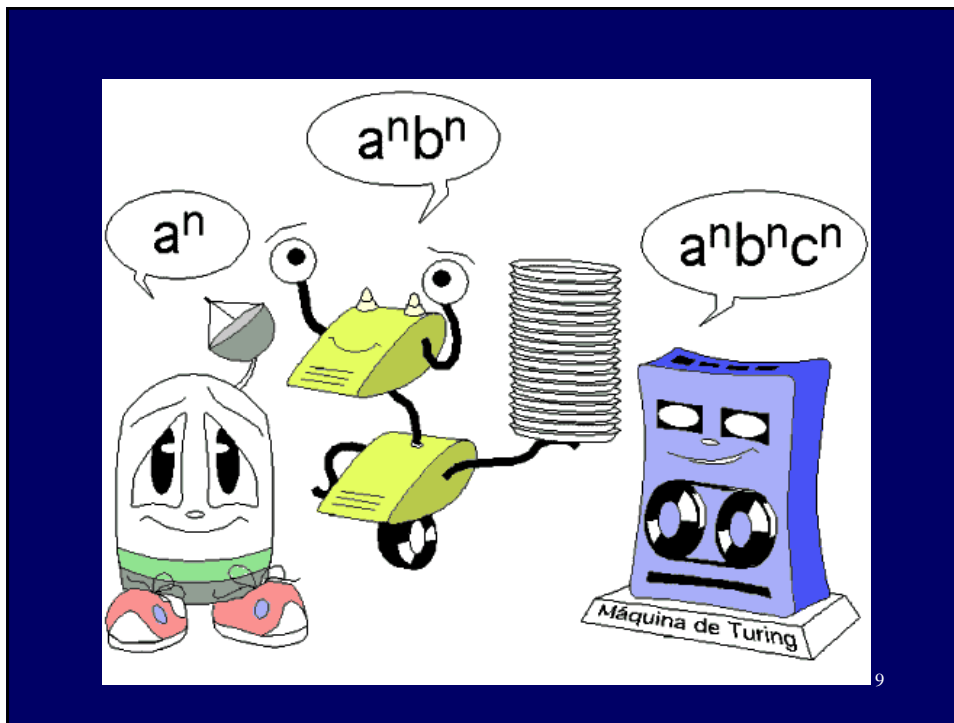
Como a história inteira não pode ser lembrada temos que escolher o que é importante esquecer e o que não é!

7

The screenshot shows the JFLAP interface in simulation mode. The finite automaton diagram is the same as in slide 7, but the transitions are now labeled "ligar" and "desligar" without quotes. Below the diagram is a simulation window with a text input field containing "ligar desligar ligar". At the bottom, there are control buttons: Step, Reset, Freeze, Thaw, Trace, and Remove.

Transições sem aspas - entrada fica confusa

8



Linguagens Abstratas?
Relação das Linguagens Formais com a
Definição de LPs

1) $L1 = \{a^n b^n \mid n \geq 0\}$

2) $L2 = \{w c w \mid w \in \{a, b\}^*\}$ ex: aabcaab

3) $L3 = \{a^n b^m c^n d^m \mid n \geq 1 \text{ e } m \geq 1\}$

$$1) L1 = \{a^n b^n \mid n \geq 0\}$$

Esta linguagem é livre de contexto e é análoga às linguagens que permitem duplo balanceamento:

- a) Linguagens bloco-estruturadas do tipo $BEGIN^n END^n$
- b) Linguagens com parênteses balanceados na forma $(^n)^n$

11

$$2) L2 = \{wcw \mid w \in \{a,b\}^*\}$$

ex: aabcaab

Esta ling não é livre de contexto e ela abstrai o problema de **checar se os identificadores foram declarados antes de serem usados no corpo de um programa Pascal**, por exemplo.

12

$$3) L3 = \{a^n b^m c^n d^m \mid n \geq 1 \text{ e } m \geq 1\}$$

Esta linguagem não é livre de contexto e ela abstrai o problema de **checar se o número de parâmetros formais na declaração de um procedimento concorda com o número de parâmetros reais no uso do procedimento.**

$a^n b^m$ poderiam representar a lista de par. formais em 2 proc. declarados com n e m argumentos e $c^n d^m$ a lista de par reais na chamada destes 2 procedimentos.

13

Aplicações

- **AF/Linguagens Regulares/Expressões Regulares:**
 - Analisador léxico (Scanner) de um **compilador**
 - Casamento de padrões em utilitários de busca
 - Verificação de protocolos de comunicação ou protocolos para troca segura de informação na Internet
 - Projeto e verificação de circuitos digitais
- **APilha/Linguagens Livre de Contexto**
 - Analisador sintático (Parser) de um **compilador**
 - Linguagens de descrição de documentos (DTD's ou schemas) como SGML, XML.

14

Estrutura de um Compilador

