LFA e Máquina de Turing

Esdras Lins Bispo Jr.
bispojr@ufg.br

Teoria da Computação
Bacharelado em Ciência da Computação

23 de abril de 2014

UFG
Campus Jataí

Plano de Aula

1. Pensamento
2. Avisos
3. Revisão
   - Autômato Finito Não-Determinístico
4. LFA
   - Expressões Regulares
5. Máquinas de Turing

Sumário

UFG
Campus Jataí

Pensamento

Pensamento

Frase

A moderação e a coragem, portanto, são destruídas pela deficiência e pelo excesso e preservadas pelo meio termo.

Quem?

Aristóteles (384 a.C. - 322 a.C.)
Filósofo e lógico grego.

UFG
Campus Jataí

Sumário

UFG
Campus Jataí

Avisos

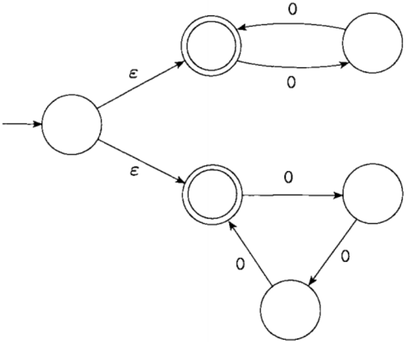
Questão Avaliada 02 no Canvas

Devo disponibilizá-la novamente!!!

Sumário

Autômatos Finitos Não-Determinístico

Qual linguagem este AFN reconhece?



Autômatos Finitos Não-Determinístico

Teorema 1.39

Todo autômato finito não-determinístico tem um autômato finito determinístico equivalente.

Autômatos Finitos Não-Determinístico

Teorema 1.39

Todo autômato finito não-determinístico tem um autômato finito determinístico equivalente.

Corolário 1.40

Uma linguagem é regular se e somente se algum autômato finito não-determinístico a reconhece.

Sumário

Expressões Regulares

Digamos que R é uma **expressão regular** (ER) se R for:

1. a, para algum $a \in \Sigma$,
2. ϵ,
3. $\emptyset$,
4. $(R_1 \cup R_2)$, em que R_1 e R_2 são expressões regulares,
5. $(R_1 \circ R_2)$, em que R_1 e R_2 são expressões regulares,
6. (R_1^*), em que R_1 é uma expressão regular.

Exemplos de ER

- 0^*10^*
- $\Sigma^*1\Sigma^*$
- $\Sigma^*001\Sigma^*$
- $1^*(01^+)^*$
- $(\Sigma\Sigma)^*$
- $(0 \cup \epsilon)1^* = 01^* \cup 1^*$
- $1^*\emptyset = \emptyset$
- $\emptyset^* = \{\epsilon\}$

Expressões Regulares

Teorema

Uma linguagem é regular se e somente se alguma expressão regular a descreve.

Expressões Regulares

Teorema

Uma linguagem é regular se e somente se alguma expressão regular a descreve.

Estratégia

Utilizar para realizar a prova um **autômato finito não-determinístico generalizado**.

Linguagens Não-Regulares

Existem linguagens que não são regulares como $A = \{0^n1^n \mid n \geq 0\}$.

Linguagens Não-Regulares

Existem linguagens que não são regulares como
$A = \{0^n 1^n \mid n \geq 0\}$.

Lema do Bombeamento

Se A é uma linguagem regular, então existe um número p (o comprimento do bombeamento) tal que, se s é qualquer cadeia de A de comprimento no mínimo p, então s pode ser dividida em três partes, $s = xyz$, satisfazendo as seguintes condições:

1. para cada $i \geq 0, xy^i z \in A$,
2. $|y| > 0$, e
3. $|xy| \leq p$.

Sumário

UFG
Campus Jataí

Modelos Básicos Computacionais

AFDs, AFNs, e Expressões Regulares

- Potencialidades: reconhecem linguagens como $(10 \cup 1)^*$;
- Fragilidades: não reconhecem linguagens como $A = \{0^n 1^n \mid n \geq 0 \text{ e } n \in \mathbb{N}\}$.

Modelos Básicos Computacionais

AFDs, AFNs, e Expressões Regulares

- Potencialidades: reconhecem linguagens como $(10 \cup 1)^*$;
- Fragilidades: não reconhecem linguagens como $A = \{0^n1^n \mid n \geq 0 \text{ e } n \in \mathbb{N}\}$.

GLCs e Autômatos com Pilha

- Potencialidades: reconhecem linguagens como $A = \{0^n1^n \mid n \geq 0 \text{ e } n \in \mathbb{N}\}$.;
- Fragilidades: não reconhecem linguagens como $A = \{a^nb^nc^n \mid n \geq 0 \text{ e } n \in \mathbb{N}\}$.

Modelos Básicos Computacionais

AFDs, AFNs, e Expressões Regulares

- Potencialidades: reconhecem linguagens como $(10 \cup 1)^*$;
- Fragilidades: não reconhecem linguagens como $A = \{0^n 1^n \mid n \geq 0 \text{ e } n \in \mathbb{N}\}$.

GLCs e Autômatos com Pilha

- Potencialidades: reconhecem linguagens como $A = \{0^n 1^n \mid n \geq 0 \text{ e } n \in \mathbb{N}\}$.;
- Fragilidades: não reconhecem linguagens como $A = \{a^n b^n c^n \mid n \geq 0 \text{ e } n \in \mathbb{N}\}$.

Portanto são bem restritos para servir de modelo de computadores de propósito geral.

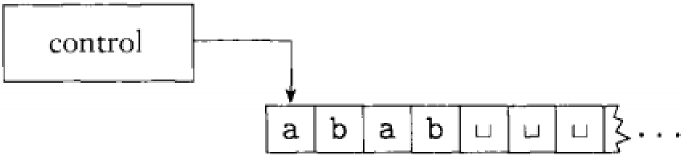
Máquinas de Turing (MT)

- Modelo mais poderoso que GLCs e AFDs;
- Turing, 1936;
- Características importantes:
    1. faz tudo o que um computador real pode fazer;
    2. existem certos problemas que uma MT não pode resolver.

Máquinas de Turing (MT)

- *Salaminh salah-mês*... tranforme as figuras em inglês!

Máquinas de Turing (MT)



Máquinas de Turing (MT)

Diferenças entre MT e AFDs

- Uma MT pode tanto escrever sobre a fita quanto ler a partir dela;
- A cabeça de leitura-escrita pode mover-se tanto para a esquerda quanto para a direita;
- A fita é infinita;
- Os estados especiais para rejeitar e aceitar fazem efeito imediatamente.

UFG
Campus Jataí

Máquinas de Turing (MT)

Construindo uma MT

Construir M_1 que reconheça a linguagem
$B = \{\omega\#\omega \mid \omega \in \{0,1\}^*\}$.

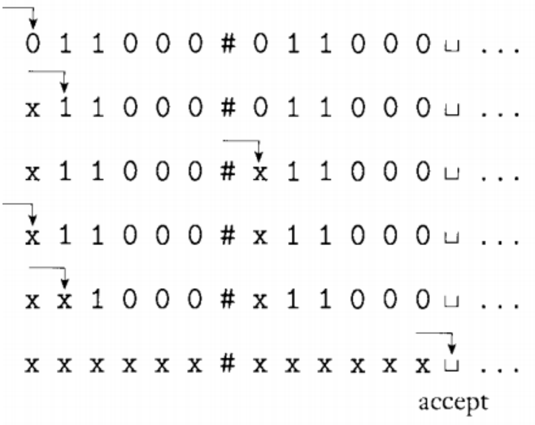
Máquinas de Turing (MT)

Descrição de M_1

M_1 = "Sobre a cadeia de entrada ω:

1. Faça um zigue-zague ao longo da fita checando posições correspondentes de ambos os lados do símbolo # para verificar se elas contêm o mesmo símbolo. Se elas não contêm, ou se nenhum # for encontrado, *rejeite*. Marque os símbolos à medida que eles são verificados para manter registro de quais símbolos têm correspondência.
2. Quando todos os símbolos à esquerda do # tiverem sido marcados, verifique a existência de algum símbolo remanecente à direta do #. Se resta algum símbolo, *rejeite*; caso contrário, *aceite*.

Máquinas de Turing (MT)



Máquinas de Turing (MT)

Uma **máquina de Turing** é uma 7-upla $(Q, \Sigma, \Gamma, \delta, q_0, q_{aceita}, q_{rejeita})$, de forma que Q, Σ, Γ são todos conjuntos finitos e

1. Q é o conjunto de estados,
2. Σ é o alfabeto de entrada sem o **símbolo branco** $\sqcup$,
3. Γ é o alfabeto da fita, em que $\sqcup \in \Gamma$ e $\Sigma \subseteq \Gamma$,
4. $\delta : Q \times \Gamma \to Q \times \Gamma \times \{E, D\}$ é a função de transição,
5. $q_0 \in Q$ é o estado inicial,
6. $q_{aceita} \in Q$ é o estado de aceitação, e
7. $q_{rejeita} \in Q$ é o estado de rejeição, em que $q_{rejeita} \neq q_{aceita}$.

Lista de Exercícios 02

Livro

SIPSER, M. **Introdução à Teoria da Computação**, 2a Edição, Editora Thomson Learning, 2011. **Código Bib.: [004 SIP/int]**.

Exercícios

- 1.4 (a, d, g);
- 1.7 (a, d, g);
- 1.15;
- 1.31.

LFA e Máquina de Turing

Esdras Lins Bispo Jr.
bispojr@ufg.br

Teoria da Computação
Bacharelado em Ciência da Computação

23 de abril de 2014

UFG
Campus Jataí