

ECM253 – Linguagens Formais, Autômatos e Compiladores

Lista de Exercícios

Máquinas de Turing

Marco Furlan

5 de junho de 2020

1 Máquinas de Turing

- 1.1. Projete uma máquina de Turing que some o valor 1 a um número N escrito em binário. Inicialmente, a fita deverá conter o símbolo # seguido pelo número N . A sua máquina de Turing deverá parar com o número $N + 1$ em binário após o símbolo #. Por exemplo, se a configuração inicial de sua máquina for $q_0\#10011$, a configuração final em um estado q_f deverá ser $q_f\#10100$ e se for $q_0\#11111$, a final deverá ser $q_f100000$. Note que você poderá remover o # mais à esquerda, se necessário.
- 1.2. Elaborar uma máquina de Turing que some dois números naturais representados em um alfabeto unário. Um número natural representado em um alfabeto unário é aquele cuja magnitude é igual ao número de repetições do símbolo utilizado pelo alfabeto. Por exemplo, utilizando o número 1, o número 5 será representado por 11111 e o número 3 por 111. Defina as condições iniciais da fita para este problema.
- 1.3. Considerando o alfabeto de entrada $\Sigma = \{a, b\}$, projetar uma máquina de Turing que aceite a linguagem $L = \{w \in \Sigma^* \mid \text{o comprimento de } w \text{ é múltiplo de } 3\}$.
- 1.4. Elaborar uma máquina de Turing que separe os símbolos de uma cadeia $w \in \Sigma^*$, onde $\Sigma = \{a, b\}$, em duas partes, por exemplo:

- Se a entrada for #abbaabaaa#, o resultado na fita deverá ser #aaaaaabb# (ou #bbbaaaaa#, se preferir);
 - Se a entrada for #bababab#, o resultado na fita deverá ser #bbbbbbaa# (ou #aaabbbb#, se preferir);
 - Se a entrada for #aaaa#, o resultado na fita deverá ser #aaa#;
 - Se a entrada for #bb#, o resultado na fita deverá ser #bb#;
 - Se a entrada for ##, o resultado na fita deverá ser ##.
- 1.5. Considerando o alfabeto de entrada $\Sigma = \{a, b\}$, projetar uma máquina de Turing que aceite a linguagem $L = \{w \in \Sigma^* \mid \text{a quantidade de símbolos } a \text{ é igual à quantidade de símbolos } b \text{ em } w\}$. Sugestão: veja se é possível aproveitar a solução do exercício anterior para facilitar a solução deste.

2 Gramáticas

- 2.1. Seja $V = \{S, A, B, a, b\}$ e $\Sigma = \{a, b\}$. Descobrir a linguagem gerada pela gramática $G = (V, \Sigma, R, S)$ quando:
- $R = \{S \rightarrow AB, S \rightarrow aA, A \rightarrow a, B \rightarrow ba\}$
 - $R = \{S \rightarrow AA, S \rightarrow B, A \rightarrow aaA, A \rightarrow aa, B \rightarrow bB, B \rightarrow b\}$
- 2.2. Seja $G = (V, \Sigma, R, S)$ uma gramática de estrutura frasal com $V = \{0, 1, A, B, S\}$, $\Sigma = \{0, 1\}$, e um conjunto de produções $R = \{S \rightarrow 0A, S \rightarrow 1A, A \rightarrow 0B, B \rightarrow 1A, B \rightarrow 1\}$.
- Mostrar que a cadeia 10101 pertence à linguagem gerada por G .
 - Mostrar que a cadeia 10110 não pertence à linguagem gerada por G .
 - Qual é a linguagem gerada por G ?
- 2.3. Elaborar uma gramática de estrutura frasal que gere a linguagem $\{0^n 1^m 0^n \mid m \geq 0 \wedge n \geq 0\}$.
- 2.4. A **notação BNF** (Backus-Naur Form) é uma alternativa popular para a especificação de gramáticas do tipo 2 (livres de contexto), por exemplo, Java. Produções em uma linguagem do tipo 2 possuem um único símbolo não terminal como lado esquerdo.

No lugar de listar todas as produções separadamente, pode-se combinar todas aquelas que possuem o mesmo símbolo terminal no lado esquerdo com uma única declaração. No lugar de utilizar o símbolo “ \rightarrow ” na produção, utiliza-se o símbolo “ $:$ ” e “ $=$ ”. Para facilitar a visualização, pode-se ladeá-los com os símbolos “ $<$ ” e “ $>$ ” e listar todos os lados direitos das produções com um lado esquerdo em comum separados pelo símbolo “ $|$ ”.

Assim as produções $A \rightarrow Aa$, $A \rightarrow a$ e $A \rightarrow AB$ podem ser combinadas como $\langle A \rangle ::= \langle A \rangle a \mid a \mid \langle A \rangle \langle B \rangle$. Por exemplo, identificadores em Algol 60 são assim especificados em BNF:

```
<identifier> ::= <letter>
| <identifier> <letter>
| <identifier> <digit>
<letter> ::= a | b | ... | y | z
<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Pede-se especificar com a notação BNF identificadores que consistem em:

- (a) Uma ou mais letras.
- (b) Pelo menos três mas não mais que seis letras minúsculas.
- (c) Uma a seis letras minúsculas ou maiúsculas, mas que comecem com uma letra maiúscula.
- (d) Uma letra minúscula seguida por um dígito ou o símbolo “_” seguido por três ou quatro caracteres alfanuméricos (letras maiúsculas ou minúsculas e dígitos).