



Universidade de Aveiro

Departamento de Electrónica, Telecomunicações e Informática

Linguagens Formais e Autómatos

Exame NM

(Ano Lectivo de 2015/16)

21 de Junho de 2016

NOTA: O exame tem 13 questões. As 10 mais bem classificadas serão cotadas a 1,6 valores cada; a mais mal classificada será cotada a 1,2 valores; as restantes 2 serão cotadas a 1,4 valores cada.

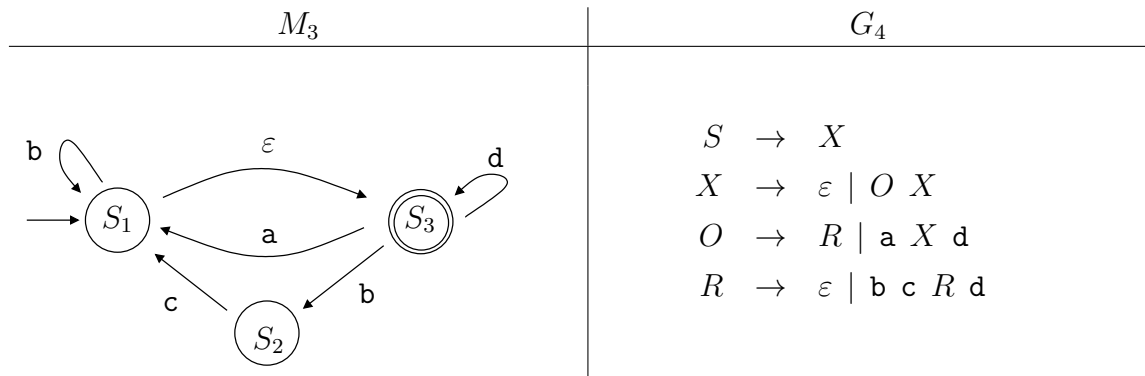
1. Considere, sobre o alfabeto $T = \{a, b, c, d\}$, as linguagens L_1 , L_2 , L_3 , e L_4 definidas da seguinte forma:

$$L_1 = \{ a^{2n} (bc)^k d^{2n+k} : n \geq 0 \wedge k \geq 0 \}$$

$$L_2 = \{ w \in T^* : w \text{ é gerada pela expressão regular } e_2 = a^*(cb)^*(a|d)^+ \}$$

$$L_3 = \{ w \in T^* : w \text{ é reconhecida pelo autómato } M_3 \}$$

$$L_4 = \{ w \in T^* : w \text{ é gerada pela gramática } G_4 \}$$

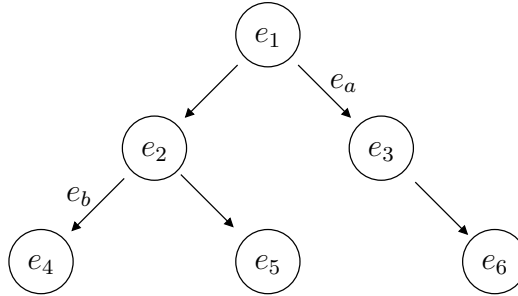


- Determine as palavras do conjunto $\{\omega \in L_1 \cap L_3 : |\omega| \leq 8\}$.
- Obtenha um **autómato finito, não generalizado**, que represente a linguagem L_2 . Apresente o raciocínio e/ou os passos intermédios usados para chegar à sua resposta.
- Obtenha um **autómato finito**, que represente a linguagem $\overline{L_3}$ (complementar de L_3). Apresente o raciocínio e/ou os passos intermédios usados para chegar à sua resposta.
- Determine uma **gramática regular** que represente a linguagem L_3 .
- Projecte uma **gramática independente do contexto** que represente a linguagem L_1 .
- Relativamente à gramática G_4 , determine o conjunto $\text{predict}(R \rightarrow \varepsilon)$. Apresente o raciocínio e/ou os passos intermédios usados para chegar à sua resposta.
- Mostre que todos os símbolos não terminais da gramática G_4 são produtivos e acessíveis.
- Mostre que $L_1 \subset L_4$.

2. Sobre o alfabeto $T = \{ e ; : () \}$, considere a gramática G dada a seguir e seja L a linguagem por ela descrita.

$$\begin{aligned} S &\rightarrow N \\ N &\rightarrow e \mid e (L) \\ L &\rightarrow A \mid L ; A \\ A &\rightarrow e : N \mid : N \end{aligned}$$

L representa grafos acíclicos (árvores) com etiquetas nos nós e com etiquetas opcionais nos arcos. Por exemplo



é uma árvore com 6 nós e 5 arcos, dos quais 3 não têm etiquetas. Esta árvore é descrita pela palavra $e_1(:e_2(e_b:e_4;:e_5);e_a:e_3(:e_6))$ em que os índices nos símbolos terminais e apenas existem para tornar clara a correspondência entre a árvore e a palavra.

Considere ainda a coleção canônica de conjuntos de itens, usada na construção de um reconhecedor (*parser*) ascendente, parcialmente apresentada a seguir e onde a função $\delta(Z_i, a)$ representa a transição de estado.

$$Z_0 = \{ S \rightarrow \bullet N, N \rightarrow \bullet e, N \rightarrow \bullet e (L) \}$$

$$Z_1 = \delta(Z_0, N) = \{ S \rightarrow N \bullet \}$$

$$Z_2 = \delta(Z_0, e) = \{ N \rightarrow e \bullet, N \rightarrow e \bullet (L) \}$$

$$Z_3 = \delta(Z_2, () = \{ N \rightarrow e (\bullet L), L \rightarrow \bullet A, L \rightarrow \bullet L ; A, A \rightarrow \bullet e : N, A \rightarrow \bullet : N \}$$

$$Z_4 = \delta(Z_3, A) = \{ L \rightarrow A \bullet \}$$

$$Z_5 = \delta(Z_3, e) = \{ \dots \}$$

$$Z_6 = \delta(Z_3, :) = \{ \dots \}$$

$$Z_7 = \delta(Z_3, L) = \{ \dots \}$$

...

- Trace a árvore de derivação da palavra $e(:e(e:e;:e);e:e(:e))$, que corresponde à palavra apresentada acima e que representa o grafo da figura.
- A gramática G não é apropriada para a construção direta de um reconhecedor (*parser*) descendente com *lookahead* de 1. Diga porquê e proponha uma gramática equivalente (que represente a mesma linguagem) que seja apropriada.
- Preencha as linhas da tabela de reconhecimento (*parsing*) para um reconhecedor ascendente relativamente aos estados Z_0 a Z_4 .
- Determine o valor dos estados (conjuntos de itens) Z_5 a Z_7 e de mais 2 à sua escolha.
- Sem alterar a gramática, construa uma gramática de atributos que associe ao símbolo inicial S um atributo que indique o número de arcos não etiquetados que a árvore contém. Introduza os atributos auxiliares que necessite.

Nos algoritmos seguintes considere uma gramática genérica $G = (T, N, S, P)$.

ALGORITMO do **first**(α), com $\alpha \in (T \cup N)^*$:

```
{
  if ( $\alpha == \varepsilon$ ) then
    return  $\{\varepsilon\}$ 
  else if ( $\alpha == a$  and  $a \in T$ ) then
    return  $\{a\}$ 
  else if ( $\alpha == B$  and  $B \in N$ ) then
     $M = \{\}$ 
    foreach ( $B \rightarrow \gamma$ )  $\in P$ 
       $M = M \cup \text{first}(\gamma)$ 
    return  $M$ 
  else /*  $|\alpha| > 1$  */
     $x = \text{head}(\alpha)$  /* the first symbol */
     $\beta = \text{tail}(\alpha)$  /* all but the first symbol */
     $M = \text{first}(x)$ 
    if  $\varepsilon \notin M$  then
      return  $M$ 
    else
      return  $(M - \{\varepsilon\}) \cup \text{first}(\beta)$ 
}
```

ALGORITMO do **follow**:

- (a) $\$ \in \text{follow}(S)$.
- (b) se $(A \rightarrow \alpha B) \in P$, então $\text{follow}(B) \supseteq \text{follow}(A)$.
- (c) se $(A \rightarrow \alpha B \beta) \in P$ e $\varepsilon \notin \text{first}(\beta)$, então $\text{follow}(B) \supseteq \text{first}(\beta)$.
- (d) se $(A \rightarrow \alpha B \beta) \in P$ e $\varepsilon \in \text{first}(\beta)$, então $\text{follow}(B) \supseteq ((\text{first}(\beta) - \{\varepsilon\}) \cup \text{follow}(A))$.

ALGORITMO do **predict**:

$$\text{predict}(A \rightarrow \alpha) = \begin{cases} \text{first}(\alpha) & \varepsilon \notin \text{first}(\alpha) \\ (\text{first}(\alpha) - \{\varepsilon\}) \cup \text{follow}(A) & \varepsilon \in \text{first}(\alpha) \end{cases}$$
