



Universidade de Aveiro

Departamento de Electrónica, Telecomunicações e Informática

Linguagens Formais e Autómatos

Exame

(Ano Lectivo de 2008/9)

18 de Junho de 2009

Sobre o alfabeto $T_1 = \{\text{m t}, = \text{e}; \text{id}\}$ considere a gramática G_1 dada a seguir e seja L_1 a linguagem por ela descrita.

$$\begin{aligned} S &\rightarrow \lambda \mid D ; S \\ D &\rightarrow \text{t } L \mid \text{m t } L \\ L &\rightarrow \text{id } X Z \\ X &\rightarrow \lambda \mid = \text{e} \\ Z &\rightarrow \lambda \mid , L \end{aligned}$$

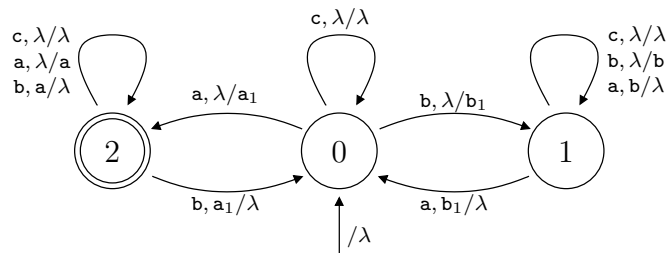
1. Relativamente à gramática G_1 :

- [1,2] (a) Usando uma derivação à esquerda, mostre que “ $\text{t id}, \text{id} = \text{e};$ ” $\in L_1$.
- [1,2] (b) Para além da regra “ $S \rightarrow D ; S$ ” a gramática G_1 tem outro ponto de recursividade à direita. Diga qual, justificando porquê.
- [1,2] (c) Mostre que todos os símbolos não terminais são produtivos.
- [1,2] (d) Mostre que $\{\lambda, ,\} \subseteq \text{first}(XZ)$.
- [1,2] (e) Mostre que $;\in \text{follow}(X)$.

[2,0] 2. Determine uma gramática equivalente a G_1 que não possua recursividade à direita.

[2,5] 3. Construa a tabela de *parsing* para um reconhecedor (*parser*) descendente com *lookahead* de 1 da linguagem L_1 . Sabe-se que, relativamente à gramática G_1 , $\text{follow}(X) = \{, ;\}$, $\text{follow}(Z) = \{;\}$ e $\text{follow}(S) = \{\$\}$,

Considere o autómato de pilha da figura abaixo, definido sobre o alfabeto $T_2 = \{\text{a b c}\}$, tendo como critério de aceitação o terminar no estado 2, qualquer que seja o conteúdo da pilha. Seja L_2 a linguagem reconhecida pelo autómato M_2 .



[1,5] 4. Mostre, usando uma sequência de configurações, que a palavra **cabaca** pertence a L_2 . Relembro que uma configuração é um triplo $(q_i, \gamma_i, u_i) \in Q \times Z^* \times T^*$, onde Q é o conjunto de estados, Z o alfabeto da pilha e T o alfabeto de entrada, que representa o estado global do autómato num dado momento no processo de reconhecimento. O processo de reconhecimento corresponde a fazer avançar o autómato de configuração em configuração desde a inicial até uma final.

continua no verso

- [3,0] 5. Construa uma gramática independente do contexto que represente a linguagem L_2 . Sabe-se que:

- O autômato M_2 representa o conjunto das palavras, definidas sobre o alfabeto T_2 , com um número de símbolos **a** superior ao número de símbolos **b**.
- É possível dividir-se qualquer palavra pertencente a L_2 em duas sub-palavras tal que:
 - a sub-palavra da esquerda tem igual número de símbolos **a** e **b**;
 - a sub-palavra da direita corresponde à concatenação de uma ou mais sub-palavras, cada uma delas com um número de símbolos **a** igual ao número de símbolos **b** mais um.

Sobre o alfabeto $T_3 = \{\mathbf{m} \ \mathbf{t} \ , \ = \ ; \ \mathbf{id}\}$ considere a gramática G_3 dada a seguir e seja L_3 a linguagem por ela descrita.

$$\begin{aligned} S' &\rightarrow S \\ S &\rightarrow \lambda \mid D \ ; \ S \\ D &\rightarrow \mathbf{t} \ L \mid \mathbf{m} \ \mathbf{t} \ L \\ L &\rightarrow V \mid V \ , \ L \\ V &\rightarrow \mathbf{id} \mid \mathbf{id} = e \end{aligned}$$

- [1,5] 6. A palavra “ $\mathbf{t} \ \mathbf{id} \ ; \ \mathbf{m} \ \mathbf{t} \ \mathbf{id} \ , \ \mathbf{id} = e \ ;$ ” pertence a L_3 . Trace a sua árvore de derivação.
- [2,0] 7. A construção de um reconhecedor (*parser*) ascendente para uma gramática baseia-se na colecção (canónica) de conjuntos de itens. O elemento inicial dessa colecção para a gramática G_3 está parcialmente descrito a seguir.

$$I_0 = \{S' \rightarrow \cdot S\} \cup \dots$$

Complete-o.

- [2,5] 8. A gramática G_3 representa uma abstracção simplificada de uma declaração de variáveis em C. O terminal **t** representa o tipo (**int**, **double**, ...) e o terminal **m** um modificador (por exemplo o **static**). Sabendo que:

- O símbolo terminal **t** tem um atributo **t** que representa o tipo que lhe está associado.
- O símbolo terminal **id** tem um atributo **v** que representa o identificador que lhe está associado.
- se dispõe de uma função de manipulação de uma tabela de identificadores, com a assinatura **addsym(x, y, z)**, que permite adicionar um elemento à tabela, onde:
 - **x** representa o identificador (valor do atributo **v** de um **id**);
 - **y** representa o tipo (valor do atributo **t** do terminal **t** que se aplica ao **id**;
 - **z** é um valor booleano que indica se o **id** é afectado ou não por um **m**

construa uma gramática de atributos que permita invocar a função **addsym** de forma adequada por cada **id** declarado.