



Universidade de Aveiro

Departamento de Electrónica, Telecomunicações e Informática

Linguagens Formais e Autómatos

Exame

(Ano Lectivo de 2012/13)

27 de Junho de 2013

1. Sobre o alfabeto $T_1 = \{a \ b \ c \ d \ e\}$ considere a gramática G_1 dada a seguir e seja L_1 a linguagem por ela descrita.

$$\begin{aligned} S &\rightarrow A \ B \\ A &\rightarrow a \mid a \ C \mid C \ a \\ B &\rightarrow b \ B \ A \mid E \\ C &\rightarrow c \ A \mid A \ c \\ D &\rightarrow d \ D \mid d \ B \\ E &\rightarrow \lambda \mid e \ E \end{aligned}$$

Avalie a veracidade das seguintes afirmações, justificando adequadamente cada uma das suas respostas.

- [1,5] (a) $acacae \in L_1$.
[1,5] (b) D é um símbolo útil. (Nota: um símbolo diz-se útil se for simultaneamente produtivo e acessível.)
[1,5] (c) $c \in \text{first}(B \ A)$.
[1,5] (d) $\lambda \in \text{first}(B \ A)$.
[1,5] (e) $\{a, \$\} \subseteq \text{follow}(A)$. (Nota: \subseteq significa subconjunto em sentido lato.)
[1,5] (f) A palavra aca permite mostrar que a gramática G_1 é ambígua.
[1,5] (g) G_1 é inadequada à implementação de um reconhecedor descendente porque possui recursividade à esquerda.

-
2. Sobre o alfabeto $T_2 = \{c \ v \ t \ d \ s \ i \ = \ e\}$ considere a gramática G_2 dada a seguir e seja L_2 a linguagem por ela descrita.

$$\begin{aligned} S &\rightarrow D \\ D &\rightarrow C \ t \ L \ d \\ C &\rightarrow c \mid v \mid \lambda \\ L &\rightarrow i \ = \ e \ X \\ X &\rightarrow \lambda \mid s \ L \end{aligned}$$

- [1,5] (a) Trace a árvore de derivação da palavra " $t \ i \ = \ e \ s \ i \ = \ e \ d$ " $\in L_2$.
[2,0] (b) Construa a tabela de reconhecimento (*parsing*) para um reconhecedor descendente.
[2,0] (c) A construção de um reconhecedor (*parser*) ascendente para uma gramática baseia-se na coleção (canónica) de conjuntos de itens. O elemento inicial dessa coleção para a gramática G_2 está parcialmente descrito a seguir.

$$Z_0 = \{S \rightarrow \cdot D\} \cup \dots$$

Complete-o e determine também os elementos diretamente alcançáveis a partir dele.

[2,0] (d) A gramática G_2 representa uma abstração de uma declaração de constantes e de variáveis inicializadas. Sabendo que:

- o símbolo terminal **t** possui um atributo chamado **type** que representa o tipo específico que lhe está associado.
- o símbolo terminal **i** tem um atributo chamado **name** que representa o nome da constante ou variável que lhe está associado.
- o símbolo terminal **e** tem um atributo chamado **value** que representa uma grandeza numérica.
- se dispõe de uma função de manipulação de uma tabela de símbolos para inserções de novas entradas, com a assinatura **addsym(c, n, t, v)**, onde
 - **c** é um parâmetro booleano, que indica se se trata da inserção de uma constante ou de uma variável;
 - **n** representa o nome da variável ou constante;
 - **t** representa o tipo específico;
 - **v** representa o valor a atribuir à constante ou variável.

construa uma gramática de atributos que permita invocar a função **addsym** de forma adequada por cada constante ou variável declarada.

[2,0] 3. Considere que processa um texto representando código C sintaticamente correto, removendo-lhe todos os caracteres com exceção das chavetas usadas para definir blocos e dos parêntesis retos usados para definir e aceder a arrays. Cada texto representando código C dá origem a uma palavra definida sobre o alfabeto $T_3 = \{\{ \} []\}$. Por exemplo o código seguinte

```
double X[10][20];
int main(void)
{
    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 20; j++)
            X[i][j] = f(i, j);
    }
    return 0;
}
double f(int i, int j)
{
    return i*10 + j;
}
```

dá origem à palavra “[] [] { { [] [] } } { }”. O conjunto das palavras assim definidas representa uma linguagem sobre o alfabeto T_3 . Seja L_3 essa linguagem. Considere que não podem aparecer chavetas dentro de parêntesis retos. Projecte um autómato de pilha que reconheça a linguagem L_3 .
