



# Universidade de Aveiro

Departamento de Electrónica, Telecomunicações e Informática

## Linguagens Formais e Autómatos

Exame

(Ano Lectivo de 2011/12)

19 de Junho de 2012

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

$$\begin{aligned} S &\rightarrow X \ a \ Y \ Z \ b \\ X &\rightarrow Y \ Z \mid c \ X \ d \\ Y &\rightarrow e \ Y \mid \lambda \\ Z &\rightarrow X \ f \mid \lambda \end{aligned}$$

- [ 2,0 ] (a) Trace a árvore de derivação da palavra “cefdab”  $\in L_1$ .
- [ 2,0 ] (b) Determine o conjunto dos símbolos não terminais produtivos de  $G_1$ . Apresente o raciocínio e/ou cálculos usados para chegar à sua resposta.
- [ 2,0 ] (c) Determine o conjunto  $\text{follow}(Z)$ . Apresente o raciocínio e/ou cálculos usados para chegar à sua resposta.
- [ 2,0 ] (d) Determine o conjunto  $\text{first}(XaYZb)$ . Apresente o raciocínio e/ou cálculos usados para chegar à sua resposta.
- [ 2,0 ] (e) A derivação  $X \Rightarrow Y \ Z \Rightarrow Z \Rightarrow X \ f$  mostra que a gramática possui recursividade à esquerda. Altere-a de modo a obter uma equivalente sem recursividade à esquerda. Apresente o raciocínio e/ou cálculos usados para chegar à sua resposta.

- 
- [ 2,5 ] 2. 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  $T_2 = \{\{ \} [ ]\}$ . 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  $T_2$ . Seja  $L_2$  essa linguagem. Projecte uma gramática independente do contexto que represente a linguagem  $L_2$ .

- 
3. Sobre o alfabeto  $T_3 = \{\mathbf{p} \ \mathbf{n} \ \mathbf{z}\}$  considere a gramática  $G_3$  dada a seguir e seja  $L_3$  a linguagem por ela descrita.

$$\begin{aligned} D &\rightarrow S \\ S &\rightarrow \lambda \mid L S \\ L &\rightarrow \mathbf{p} P P O T \\ P &\rightarrow \mathbf{n} \ \mathbf{n} \\ O &\rightarrow \lambda \mid P O \\ T &\rightarrow \mathbf{z} \mid \lambda \end{aligned}$$

- [ 2,5 ] (a) Construa a tabela de *parsing* para um reconhecedor (*parser*) descendente com *lookahead* de 1 da linguagem  $L_3$ .
- [ 2,5 ] (b) 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.

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

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

- [ 2,5 ] (c) A gramática  $G_3$  representa um desenho constituído por uma sequência de zero ou mais *polylines* definidas num plano. Cada *polyline*, representada pelo símbolo não terminal  $L$ , é definida por dois ou mais pontos e um terminador opcional ( $\mathbf{z}$ ) que, quando presente, fecha a *polyline*. Um ponto, representado pelo símbolo não terminal  $P$ , é definido pelas suas coordenadas  $X$  e  $Y$ . Considerando que:

- o símbolo terminal  $\mathbf{n}$  representa um número e tem um atributo  $v$  que indica o valor do número,
- dispõe de uma função **drawLine**( $x_1, y_1, x_2, y_2$ ) que desenha uma linha do ponto  $(x_1, y_1)$  ao ponto  $(x_2, y_2)$

projete uma gramática de atributos que desenhe as *polylines* definidas pelas palavras da gramática. Introduza os atributos auxiliares que ache necessários.