

## Linguagens e Programação

### Exame Época Normal

20 de Julho de 2016

- Responda a cada grupo em folhas separadas
- Indique o seu número e nome em cada folha que entregar
- A prova é com consulta (1 folha A4) e tem a duração de 2 horas

#### Grupo I (6 valores)

1. [1,5 val.] Identifique as características das gramáticas de atributos e realce o seu papel na fase do processo de compilação em que são utilizadas.
2. [1 val.] “O código intermédio é independente da máquina alvo”. Comente a afirmação.
3. [3,5 val.] Considere a seguinte gramática G:  
$$C \rightarrow A b B \mid A c$$
$$B \rightarrow B b B c \mid C c$$
$$A \rightarrow a A b \mid a$$
  - a) [0,5 val.] Caracterize formalmente a gramática G.
  - b) [1,5 val.] Distinga as diferentes formas de derivação canónica seguidas para a sistematização da aplicação das regras de uma gramática no âmbito da análise sintática. Ilustre, apresentando as sequências de derivação resultantes para a frase **aabbacc**, na gramática G.
  - c) [1,5 val.] A gramática G é LL(1)? Justifique.

#### Grupo II (4 valores)

1. Para garantir a rastreabilidade dos produtos fabricados numa determinada unidade industrial, cada lote de fabrico deverá ser codificado de forma a representar a informação necessária. Os campos que devem ser definidos nesta codificação são os seguintes:

Campo	Tamanho	Descrição
Identificação do lote	5	2 dígitos para o ano seguidos de 3 para o ID antecidos dos caracteres LT
Referência da peça	4	dígitos alfanuméricos
Identificação da máquina	2	2 dígitos
Quantidade	5	5 dígitos
Data (Mês/Dia)	4	2 dígitos para o mês (01-12) seguidos de 2 para o dia (01-31)

A ordem a seguir deve ser a definida na tabela anterior e deve ser usado o hífen como separador.  
Ver exemplo seguinte:

**LT16121-3z1a-23-13450-1205**

- [1,5 val.] Escreva uma Expressão Regular que permita representar a codificação dos lotes de fabrico.
- [1 val.] Construa um AF que represente o componente **Data** da codificação descrita na alínea a). Assuma que todos os meses podem ter 31 dias. Classifique o AF obtido. Justifique.

2. [1,5 val.] Considere o autómato finito A, descrito na tabela seguinte:

	x	y
$\rightarrow s_0$	$\{s_0, s_1\}$	$\{s_0\}$
$s_1$	$\{s_1\}$	$\{s_2\}$
$*s_2$	$\emptyset$	$\{s_2\}$

Minimize o autómato finito A, **usando o método formal de minimização**. Represente graficamente o autómato finito obtido.

### Grupo III (4 valores)

1. Considere a seguinte gramática:

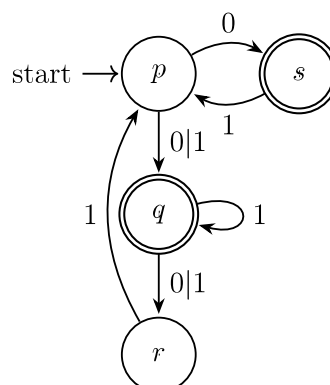
**S**  $\rightarrow$  **STS** | **L**

**L**  $\rightarrow$  **yL** | **xL** | **z**

**T**  $\rightarrow$  **-** | **#**

- [1 val.] Classifique a gramática segundo a hierarquia de Chomsky. Justifique,
- [1 val.] Demonstre que esta gramática é ambígua. Justifique.
- [1 val.] Apresente uma sequência de derivação para a palavra **yyyz#xxxxz**

2. [1 val.] Apresente uma gramática equivalente ao seguinte autómato finito:



## Grupo IV (6 valores)

Considere o seguinte extrato de um ficheiro XML:

```
<Gestao>
  <Projetos>
    <Projeto id="p001" designacao="Projeto 1">
      <!-- id é uma string, cujo formato é o carater p seguido de 3 algarismos -->
      <!-- designacao é uma string com o comprimento minimo de 5 e maximo de 30
carateres -->
      <!-- DataInicio contém uma data no formato AAAA-MM-DD -->
      <Elementos>
        <Elemento id="ps001"/>
        <!-- id é uma referência ao identificador id de Pessoas/Pessoa -->
        <Elemento id="ps002"/>
        <Elemento id="ps004"/>
      </Elementos>
      <DataInicio>2016-02-27</DataInicio>
    </Projeto>
    <Projeto id="p002" designacao="Projeto 2">
      <Elementos>
        <Elemento id="ps002"/>
        <Elemento id="ps003"/>
        <Elemento id="ps001"/>
        <Elemento id="ps004"/>
      </Elementos>
      <DataInicio>2016-01-19</DataInicio>
    </Projeto>
  </Projetos>
  <Pessoas>
    <Pessoa id="ps001" nome="Maria F." classificacao="Excelente"/>
    <!-- id é um identificador, cujo formato é "ps" seguido de 3 algarismos -->
    <!-- classificacao é um de { Excelente, Muito Bom, Bom } e é opcional -->
    <Pessoa id="ps002" nome="Carla M." classificacao="Bom"/>
    <Pessoa id="ps003" nome="Antonio C." />
    <Pessoa id="ps004" nome="Joao L." classificacao="Muito Bom"/>
    <Pessoa id="ps004" nome="Anita V." classificacao="Excelente"/>
  </Pessoas>
</Gestao>
```

1. [3 val.] Crie os tipos complexos **TProjetos** e **TPessoa** (e outros que necessite) que validem respetivamente os elementos **Projetos** e **Pessoa**, obedecendo às restrições que se encontram a comentário no extrato de XML.

```
<?xml version="1.0"?>
<xs:schema version="1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <xs:element name="Gestao" type="TGestao"/>
  <xs:complexType name="TGestao">
    <xs:sequence>
      <xs:element name="Projetos" type="TProjetos"/>
      <xs:element name="Pessoas" type="TPessoas"/>
    </xs:sequence>
  </xs:complexType>

  <!-- COMPLETAR schema -->

</xs:schema>
```

2. [3 val.] Complete a seguinte transformação XSLT (**Blocos A e B**), de modo que permita listar, por ordem crescente do seu nome, todas as pessoas envolvidas em pelo menos um projeto.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="html"/>

  <xsl:template match="/">
    <html>
      <head>
        <title>projetos.xsl</title>
      </head>
      <body>

        <!-- COMPLETAR Bloco A -->

      </body>
    </html>
  </xsl:template>

  <!-- COMPLETAR Bloco B -->

</xsl:stylesheet>
```

A legenda da tabela deverá apresentar o **número de projetos com mais de 3 pessoas.**, como apresentado na figura seguinte:

Projetos 3+: 1

Nome	Projetos
Antonio C.	1
Carla M.	2
Joao L.	2
Maria F.	2