

Linguagens e Programação

Exame Época Especial

20 de novembro de 2019

- 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 val.] Identifique e descreva a metodologia de análise sintática seguida pelo ANTLR.
2. [2 val.] Considere a gramática $G = (\{A, B\}, \{0, 1\}, \{A \rightarrow A0|B|AB, B \rightarrow 1|1B\}, A)$:
 - a) Distinga as diferentes formas de derivação canónica seguidas na sistematização da aplicação das regras de uma gramática no âmbito da análise sintática. Ilustre, apresentando as sequências de derivação resultantes para a frase **1000001**.
 - b) Analise a gramática G e indique as transformações necessárias para que a linguagem correspondente possa ser reconhecida por um analisador sintático preditivo.
3. [1,5 val.] As ideias de Alan Turing foram consideradas demasiado avançadas para a tecnologia da sua época. Saliente as principais contribuições à Computação Moderna transmitidas pela sua obra.
4. [1,5 val.] Realce as vantagens do modelo *FrontEnd-BackEnd* de um compilador.

Grupo II (4 valores)

1. Considere o autómato finito A , descrito na tabela seguinte:

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

- a) [0,75 val.] Classifique o autómato finito. Justifique.
 - b) [0,75 val.] Indique uma palavra de $\{x, y\}^*$ que pertence a $L(A)$.
 - c) [1,5 val.] Minimize o autómato finito A , usando o método formal de minimização. Represente graficamente o autómato finito obtido.
2. [1 val.] Converta a expressão regular $(0|1)^*0+01^*$ numa gramática.

Grupo III (4 valores)

1. Considere a gramática seguinte em que,

$X \rightarrow aXY|bY|Z$

$Y \rightarrow bY|Z$

$Z \rightarrow Ya|\varepsilon$

- [1 val.] Defina formalmente a gramática anterior.
- [1 val.] Classifique a gramática segundo a hierarquia de Chomsky. Justifique.
- [1 val.] Considere a palavra **ababb**, valide se pertence à linguagem gerada pela gramática. Apresente a sequência de derivação canónica mais à esquerda.
- [1 val.] Valide, justificando, a ambiguidade da gramática.

Grupo IV (6 valores)

Considere o seguinte extracto de um ficheiro XML, que contém a informação recolhida por uma agência de viagens:

```
<agencia>
  <viagens>
    <!-- origem e destino são códigos iata relativos aos aeroportos -->
    <!-- iata é o código do aeroporto restrito a OPO, LIS, FAO, MAD, PAR e LON-->
    <!-- duracao é tempo no formato hh:mm:ss -->
    <!-- custo é um valor monetário -->
    <viagem origem="OPO" destino="LIS">
      <duracao>00:45:00</duracao> <custo>40.00</custo>
    </viagem>
    <viagem origem="OPO" destino="FAO">
      <duracao>01:10:00</duracao> <custo>45.00</custo>
    </viagem>
    <viagem origem="OPO" destino="PAR">
      <duracao>02:15:00</duracao> <custo>55.00</custo>
    </viagem>
    <viagem origem="OPO" destino="LON">
      <duracao>02:30:00</duracao> <custo>49.00</custo>
    </viagem>
    <viagem origem="LIS" destino="LON">
      <duracao>02:45:00</duracao> <custo>35.00</custo>
    </viagem>
  </viagens>
  <aeroportos>
    <aeroporto>
      <!--designacao: string com comprimento máximo de 30 -->
      <iata>LIS</iata> <designacao>Lisboa</designacao>
    </aeroporto>
    <aeroporto>
      <iata>FAO</iata> <designacao>Faro</designacao>
    </aeroporto>
    <aeroporto>
      <iata>MAD</iata> <designacao>Madrid</designacao>
    </aeroporto>
    <aeroporto>
      <iata>PAR</iata> <designacao>Paris</designacao>
    </aeroporto>
    <aeroporto>
      <iata>LON</iata> <designacao>Londres</designacao>
    </aeroporto>
  </aeroportos>
</agencia>
```

1. [3 val.] Crie os tipos complexos **TViagens** e **TAeroportos** (e outros que necessite) que validem respetivamente os elementos <viagens> e <aeroportos>, obedecendo às restrições que se encontram em comentário no extrato de XML.

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

  <xsd:element name="agencia" type="TAgencia"/>
  <xsd:complexType name="TAgencia">
    <xsd:sequence>
      <xsd:element name="viagens" type="TViagens"/>
      <xsd:element name="aeroportos" type="TAeroportos"/>
    </xsd:sequence>
  </xsd:complexType>

  <!-- COMPLETAR! -->

</xsd:schema>
```

2. [3 val.] Crie uma transformação XSLT que permita listar todos os voos, com um custo inferior a 50€, que tenham como origem um dado aeroporto passado por parâmetro, ordenados de forma descendente por duração. O parâmetro contém o código **iata** de um aeroporto existente. O resultado deverá ser como o apresentado na figura abaixo, onde o parâmetro da origem toma o valor 'OPO', conforme o seguinte excerto de código que deve completar (bloco A):

```
<?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:param name="origem" select="'OPO'"/>

  <!-- bloco A -->

</xsl:stylesheet>
```

Viagens de OPO com custo inferior a 50€

Destino	Duração	Custo
Londres	02:30:00	€49.00
Faro	01:10:00	€45.00
Lisboa	00:45:00	€40.00
Totais	-	€134.00