

Linguagens e Programação

Exame Modelo

- Responda a **cada grupo em folhas separadas**. Deve entregar, pelo menos, **uma folha por grupo**
- 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.] Distinga as diferentes metodologias usadas na construção das tabelas de parse, no âmbito da análise sintática ascendente.
2. [1,5 val.] “Diz-se que uma gramática é ambígua se existirem, pelo menos, duas sequências de derivação diferentes para a mesma frase”. Comente a afirmação.
3. [1,5 val.] Um dos modelos para a construção de compiladores faz a separação entre o *Front-End*, encarregado da fase de análise, e o *Back-End*, encarregado da geração de código. Clarifique a função da tabela de símbolos no âmbito deste modelo.
4. [1,5 val.] “Numa definição dirigida pela sintaxe os atributos dos símbolos não terminais são somente sintetizados e fornecidos pelo analisador léxico”. Comente a afirmação.

Grupo II (4 valores)

1. [1 val.] Pretende-se o estudo e especificação de um sistema de codificação para produtos biológicos. Suponha que dispõe de um alfabeto com 3 caracteres $\Sigma=\{g, h, i\}$ para reconhecer identificadores de produtos. Considere que: um identificador só pode ser inicializado por **g** e este só poderá ocorrer no máximo uma vez. Defina uma expressão regular que lhe permita validar identificadores de produtos. Enuncie todos os identificadores de comprimento 2.
2. Considere o autómato finito A representado na Tabela 1:

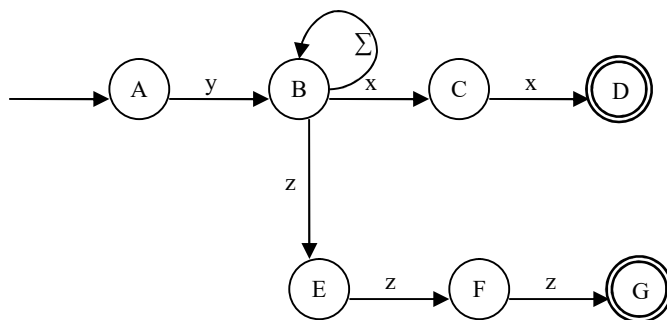
Tabela 1

	0	1
→S	{A}	{B}
A	{B}	∅
B	{C}	{B}
C	{D}	∅
*D	{D}	{C, D}

- a) [0,5 val.] Classifique o autómato finito A. Justifique.
- b) [1,5 val.] Minimize o autómato finito A, usando o método formal de minimização. Justifique todas as decisões e pressupostos que assumir. Represente graficamente o autómato finito obtido.
- c) [1 val.] Indique uma expressão regular que reconhece a linguagem aceite pelo autómato finito A.

Grupo III (4 valores)

1. Considere, no alfabeto $\Sigma = \{x, y, z\}$, o autómato finito A:



- [1,5 val.] Converta o autómato A numa gramática, usando o método formal de conversão.
- [0,5 val.] Classifique a gramática obtida na alínea anterior, segundo a hierarquia de Chomsky. Justifique.

2. Considere a gramática G:

$S \rightarrow aS \mid bX \mid cY$

$X \rightarrow aS \mid bX \mid \varepsilon$

$Y \rightarrow cZ$

$Z \rightarrow \varepsilon$

- [1,5 val.] Crie, se possível, um autómato finito equivalente a esta gramática, usando o método formal de conversão.
- [0,5 val.] Caracterize formalmente a gramática G.

Grupo IV (6 valores)

Considerando a atual conjuntura de confinamento social, o ginásio **BoraLá a Mexer** solicitou o desenvolvimento de um analisador sintático que validasse o formato dos registos das atividades dos seus clientes, para analisar o esforço diário envolvido em atividades físicas.

Para tal, desenvolveu um processo automático que permite receber a listagem com a descrição dos registos que os clientes realizaram, num ficheiro de texto cujas linhas têm a seguinte estrutura em formato EBNF (em que os campos opcionais são apresentados entre `[]` e os que se repetem 0 ou mais vezes entre `{ }`):

```
{ <tipo_de_atividade> [data] <duracao> <distancia> {marco} '\n' }
```

Em que:

- `<tipo_de_atividade>` pode ser **CORRIDA** ou **CICLISMO**;
- `<data>` é uma *String* no formato **ANO/MÊS/DIA**, indicativa da data da atividade, onde o ANO deve estar entre 2000 e 2099, o MÊS deve representar um número de mês válido com dois dígitos e o DIA um valor numérico, também com dois dígitos, entre 01 e 31;
- `<duracao>` representa o tempo da atividade, no formato **HORAS:MINUTOS**;
- `<distancia>` representa um valor **real** seguido do símbolo **KM** indicando a distância da atividade;
- `<marco>` é uma *String*, com o máximo de 32 caracteres e sem espaçamentos, com a descrição de um local de interesse por onde a atividade passou;

Defina a gramática para o ficheiro anteriormente descrito, e crie utilizando o **Flex** e o **Bison** um programa que:

- Reconheça a validade do ficheiro e implemente uma estratégia simples de recuperação de erros;
- Valide o input e, para cada linha, apresente o tempo total da atividade em minutos. No final da execução deverá apresentar a distância total percorrida global e por tipo de atividade, conforme exemplo seguinte:

Input	Output
CORRIDA 00:18 5.0KM	Duracao: 18 minutos, 18 total
CICLISMO 2019/12/20 01:30 50.0KM GONDOMAR GAIA	Duracao: 90 minutos, 108 total
CORRIDA 01:25 21.1KM	Duracao: 85 minutos, 193 total
CORRIDA 03:30 42.2KM PORTO GAIA	Duracao: 210 minutos, 403 total
	Distancia corrida: 68.3KM
	Distancia ciclismo: 50.0KM
	Distancia total: 118.3KM
	Ficheiro válido