

Nome: _____

Número: _____

Duração: 1h45

Versão A

Prova com consulta limitada a 1 folha (2 páginas) A4 manuscrita.

Não são permitidos meios eletrónicos (computador, telemóvel, ...).

Tentativas de fraude conduzem à anulação da prova para todos os intervenientes.

Responda a cada grupo em folhas separadas!
Coloque o seu nome completo e a versão do exame em todas as folhas!

Grupo I: [4 Val] Análise Lexical

Considere o excerto de código à direita, da linguagem MUPL.

a) Defina expressões regulares para os identificadores da linguagem (usados para nomes de variáveis e funções), para números inteiros e reais. Um identificador deve iniciar por minúscula, ter um tamanho mínimo de dois caracteres, e pode ser constituído por letras e dígitos. Um número real deve ter sempre pelo menos um dígito à esquerda e/ou direita do ponto¹.

b) Desenhe um DFA parcial para a linguagem, que permita o reconhecimento de identificadores, números inteiros e reais (e apenas destes três tokens), tal como definidos na alínea anterior.

```
def func echo() {  
    var x;  
    read x;  
    write x;  
    return;  
}  
def func manyEcho(amnt) {  
    loop {  
        amnt == 0 -> return;  
        call echo();  
        decr amnt;  
    }  
}
```

Grupo II: [6 Val] Análise Sintática (Analisadores Descendentes)

$S \rightarrow tSz$
 $S \rightarrow A$
 $A \rightarrow Av$
 $A \rightarrow B$
 $B \rightarrow xB$
 $B \rightarrow xp$

Considere a gramática à esquerda, em que as maiúsculas representam variáveis e as minúsculas os terminais da linguagem.

a) É possível implementar um analisador sintático descendente preditivo LL(1) para a gramática apresentada? Justifique.

b) Determine os conjuntos First e Follow das variáveis S, A e B.

c) Modifique a gramática, fatorizando-a à esquerda e eliminando a recursividade à esquerda.

d) Desenhe uma árvore concreta para a cadeia 'txxpvvzz'.

e) Desenhe uma possível árvore abstrata para a mesma cadeia.

f) Pretende-se implementar um analisador sintático LL(1) para reconhecimento da linguagem. Assuma a existência de um Analisador Lexical, uma variável global *token*; a função *next()*, que devolve o próximo token; e o atributo *text* de *token*, que indica qual o conteúdo do token. Apresente o (pseudo-)código correspondente à função que faça o processamento da variável S.

g) Construa a tabela sintática LL(1) para a gramática modificada e indique, justificando, se a gramática pode ser considerada LL(1). (Nota: caso não tenha feito a alínea c), contrua a tabela para a gramática original, indicando esse facto na sua resposta)

¹ Os números reais admitem zeros à esquerda da parte inteira e à direita da parte decimal

Grupo III: [5] Análise Sintática (Analisadores Ascendentes)

Considere a gramática apresentada à direita e a tabela do analisador sintático ascendente LR(0) abaixo.

$S \rightarrow A\$$	(1)
$A \rightarrow xAz$	(2)
$A \rightarrow B$	(3)
$B \rightarrow y$	(4)

Estado	Ação				GoTo	
	x	y	z	\$	A	B
s0	shift (s1)	shift (s4)	error	error	GoTo s6	GoTo s5
s1	shift (s1)	shift (s4)	error	error	GoTo s2	GoTo s5
s2	error	error	shift (s3)	error	-	-
s3	reduce (2)	reduce (2)	reduce (2)	reduce (2)	-	-
s4	reduce (4)	reduce (4)	reduce (4)	reduce (4)	-	-
s5	reduce (3)	reduce (3)	reduce (3)	reduce (3)	-	-
s6	error	error	error	accept	-	-

a) A gramática pode ser considerada LR(0)? Justifique a sua resposta.

b) Indique a sequência de estados (conteúdo da pilha de estados, conteúdo da pilha de símbolos, cadeia de entrada) para a cadeia 'xxyzz', indicando se a mesma é ou não aceita pela linguagem.

Pilha de Estados	Pilha de Símbolos	Entrada
s0		xxyzz
...
...

Considere a gramática apresentada à direita.

c) Construa a tabela do analisador sintático ascendente LR(0) para a gramática e indique, justificando, se a gramática pode ser considerada LR(0). (Mostre o DFA usado para a construção da tabela)

$S \rightarrow X\$$	(1)
$X \rightarrow aXb$	(2)
$X \rightarrow c$	(3)

Grupo IV: [5 Val] Compiladores (Miscelânea)

a) Indique as principais diferenças entre o funcionamento de um compilador e um interpretador.

b) Considere o seguinte excerto de código, e os problemas a ele associados. Indique em que fase de compilação (lexical, sintática, ou semântica) é detetado cada um dos seis erros apresentados.

(1) Falta tipo de dados do parâmetro

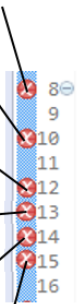
(2) '3a' não é reconhecido como identificador ou palavra reservada

(3) '**' não esperado

(4) ')' não esperado

(5) Falta ';' no final da instrução

(6) Tipo de dados retornados não corresponde ao tipo de retorno declarado



```
private static void test(a)
{
    for(i = 1 ; i < 3a ; i++)
    {
        a = a ** i;
    }
    System.out.println("a=" + a)
    return a;
}
```

Indique, **justificando sucintamente**, se cada uma das seguintes afirmações é Verdadeira ou Falsa.
(Resposta errada = desconto de 50% da cotação da alínea)

- c) As fases de análise lexical e análise sintática são normalmente implementadas conjuntamente, funcionando como um único módulo do compilador.
- d) A sequência TYPE(int) IDENTIFIER(x) IDENTIFIER(y) ATRIB INT(0) COMMA não é uma sequência de *tokens* válida enquanto saída (*output*) de um analisador lexical para uma linguagem de programação como Java.
- e) Um analisador sintático descendente preditivo pode ser usado para analisar qualquer gramática não ambígua.
- f) A definição da estrutura sintática de uma linguagem de programação como Java pode ser feita utilizando expressões regulares.
- g) Uma gramática com recursividade à esquerda não pode nunca ser processada por um analisador sintático ascendente LR(k).