

Pergunta 1

7,5 de 10 pontos

Considere a seguinte gramática que especifica uma linguagem de strings binárias com algumas restrições:

S -> '0' A
 | '1' B

A -> '0' S
 | '1' C

B -> '0' C
 | '1' S
 | ε

C -> '0' B
 | '1' A

Assinale as afirmações verdadeiras.

Respostas selecionadas: ☒ a. É possível reconhecer esta linguagem com um parser recursivo descendente.

☒ b. A string "1111100" é uma frase válida da linguagem.

☒ d. A string "10101" é uma frase válida da linguagem.

Respostas corretas: ☒ a. É possível reconhecer esta linguagem com um parser recursivo descendente.

☒ b. A string "1111100" é uma frase válida da linguagem.

O parser recursivo descendente teria a seguinte função:

```
def rec_B():  
    if prox_simb == '0':  
        rec_term('0')  
        rec_C()  
    elif prox_simb == '1':  
        rec_term('1')  
        rec_S()  
    elif prox_simb == '$':  
        pass  
    else:
```

☒ c. erro("Erro sintático em B, símbolo inesperado: " + prox_simb)

☒ d. A string "10101" é uma frase válida da linguagem.


Pergunta 2

0 de 10 pontos

Considere o seguinte programa da linguagem VM:


```
pushi 0
pushi 1
start
pushg 1
pushg 0
add
pushg 1
pushg 0
mul
sub
jz end
pushi 5
writei
end:
stop
```

Assumindo que os valores booleanos True e False são tratados como os inteiros 1 e 0, respetivamente, assinale as afirmações verdadeiras


Respostas seleccionadas:  a. Se a segunda instrução for 'pushi 0', quando executado pela VM, o programa imprime 5.

O programa é uma possível implementação do seguinte programa imperativo:

```
a = False
b = True
if b or a:
```


 d. print(5)

Respostas corretas:

 c. Quando executado pela VM, o programa imprime 5.

O programa é uma possível implementação do seguinte programa imperativo:

```
a = False
b = True
if b or a:
```

 d. print(5)

Pergunta 3

5 de 10 pontos

Considere os Terminais "str" (texto entre aspas), "texto" (sequência de caracteres) e "id" (sequência não nula de letras) e a seguinte Gramática Independente de Contexto (G):

p1: Anota -> Abre texto Fecha

p2: Abre -> '<' id '>'

p3: | '<' id LstA '>'

p4: Fecha -> '<' '/' id '>'

p5: LstA -> Atr

p6: LstA -> LstA Atr

p7: Atr -> id '=' str

Averigue então a veracidade das seguintes afirmações:

- Respostas selecionadas:
- ☒ a. O Lookahead de p1 é: $la(p1) = \{ '<' \}$
 - ☒ b. Dada a recursividade à esquerda usada na produção **p6**, não se pode calcular o seu Lookahead.
 - ☒ d. A gramática G tem um conflito LL(1) nas produções **p2** e **p3** porque os Lookahead são iguais.

- Respostas corretas:
- ☒ a. O Lookahead de p1 é: $la(p1) = \{ '<' \}$
 - ☒ d. A gramática G tem um conflito LL(1) nas produções **p2** e **p3** porque os Lookahead são iguais.

Pergunta 4

0 de 10 pontos

Considere os Terminais NInt (número inteiro), NReal (número decimal) e Pal (sequência de uma ou mais letras) e a seguinte Gramática Independente de Contexto (G):

p1: Frase -> '[' Elems ']'

p2: Elems -> €

p3: Elems -> Elem Elems

p4: Elem -> NInt

p5: | NReal

p6: | Pal

p7: | Frase

Selecione então as alíneas abaixo que são afirmações verdadeiras:

Respostas selecionadas: ☒ c. O Parser Recursivo-Descendente otimizado é composto por 4 funções de reconhecimento para os seus símbolos N e T.

☐ d. O Parser Recursivo-Descendente puro é composto por 7 funções de reconhecimento para os seus símbolos N e T.

Respostas corretas: ☒ b. É possível escrever uma função para reconhecer o símbolo 'Elem' porque não há conflito nenhum no Lookahead das produções p4 a p7.

☒ c. O Parser Recursivo-Descendente otimizado é composto por 4 funções de reconhecimento para os seus símbolos N e T.

Considere os seguintes excerto do analisador sintático, escrito em Python, e que gera código para a VM para expressões aritméticas compostas por somas, subtrações e constantes inteiras.

Preencha os espaços em branco com as ações semânticas em falta.

Nas suas respostas obedea ao seguinte:

- 1. Escreva as instruções da VM em maiúsculas;
- 2. Não usem f strings, usem a concatenação;
- 3. Utilizem aspas e não plicas;
- 4. Não coloquem "ln" nas respostas;
- 5. Assumam que todos os símbolos terminais vêm na forma de strings.

```
def p_Exp_add(p):
    "Exp : Exp '+' Termo"
    p[0] = [A]
def p_Exp_sub(p):
    "Exp : Exp '-' Termo"
    p[0] = [B]
def p_Exp_termo(p):
    "Exp : Termo"
    p[0] = [C]
def p_Termo_group(p):
    "Termo : '(' Exp ')'"
    p[0] = [D]
def p_Termo_num(p):
    "Termo : num"
    p[0] = [E]
```

- Resposta especificada para: A p[1] + p[2] + "ADD"
- Resposta especificada para: B p[1]+p[2] + "SUB"
- Resposta especificada para: C p[1]
- Resposta especificada para: D p[2]
- Resposta especificada para: E "PUSHI" + num

Respostas corretas para: A		
Método de avaliação	Resposta correta	Sensibilidade a maiúsculas e minúsculas
Correspondência de padrão	p\[1\]\s*(\+ s*"\\n")?\s* \+s*p\[3\]\s*(\+ s*"\\n")?\s* \+s*"ADD"\s*(\+ s*"\\n")?	
Respostas corretas para: B		
Método de avaliação	Resposta correta	Sensibilidade a maiúsculas e minúsculas
Correspondência de padrão	p\[1\]\s*(\+ s*"\\n")?\s* \+s*p\[3\]\s*(\+ s*"\\n")?\s* \+s*"SUB"\s*(\+ s*"\\n")?	
Respostas corretas para: C		
Método de avaliação	Resposta correta	Sensibilidade a maiúsculas e minúsculas
Correspondência de padrão	\s*p\[1\]\s*	
Respostas corretas para: D		
Método de avaliação	Resposta correta	Sensibilidade a maiúsculas e minúsculas
Correspondência de padrão	\s*p\[2\]\s*	
Respostas corretas para: E		
Método de avaliação	Resposta correta	Sensibilidade a maiúsculas e minúsculas
Correspondência de padrão	\s*"PUSHI" \s*(\+ s*p\[1\]\s*(\+ s*"\\n")?\s*	

Pergunta 6

0 de 10 pontos

Considere os Terminais "str" (texto entre aspas), "texto" (sequência de caracteres) e "id" (sequência não nula de letras) e a seguinte Gramática Independente de Contexto (G):

p1: Anota -> Abre texto Fecha

p2: Abre -> '<' id '>'

p3: | '<' id LstA '>'


p4: Fecha -> '<' '/' id '>'


p5: LstA -> Atr


p6: LstA -> LstA Atr


p7: Atr -> id '=' str

Averigue então a veracidade das seguintes afirmações:

Respostas seleccionadas:  b. Construindo o Autómato LR(0) para G verifica-se que do estado inicial há 2 transições por terminais.

 c. Construindo o Autómato LR(0) para G verifica-se que o estado inicial tem 7 itens (ou produções com marcador).

Respostas corretas:  a. Construindo o Autómato LR(0) para G verifica-se que o estado inicial tem 4 itens (ou produções com marcador).

 d. Construindo o Autómato LR(0) para G verifica-se que do estado inicial só há 3 transições por terminais ou não-terminais.

Considere uma linguagem de programação, LPIS, imperativa simples que só tem variáveis do tipo inteiro atômicas e arrays. Atente então ao fragmento de programa seguinte:

```
varint a, b
vararr v[10]
varint x=5, y=8
BEGIN
a = x+y/2
```

Selecione então as alíneas abaixo que são afirmações verdadeiras:

Respostas selecionadas: O respetivo programa Assembly da VM podia começar assim:

```
PUSHI 0
PUSHI 0
PUSHN 10
PUSHI 5
PUSHI 8
START
PUSHG 12
PUSHG 13
ADD
PUSHI 2
DIV
```

☒ c. LOAD 0

A Tabela de Identificadores do Compilador de LPIS guardaria informação como a que se mostra abaixo:

☒ d. { (a, [var, 0]), (b, [var, 1]), (v, [arr, 10, 2]), (x, [var, 12]), (a, [var, 13]) }

Respostas corretas: O respetivo programa Assembly da VM podia começar assim:

```
PUSHN 12
PUSHI 5
PUSHI 8
START
PUSHG 12
PUSHG 13
PUSHI 2
DIV
ADD
```

☒ b. STOREG 0

Pergunta 8

5 de 10 pontos

Considere a seguinte gramática G, onde os símbolos em maiúsculas são os não terminais, e em minúsculas são os terminais:

$$S \rightarrow \varepsilon$$
$$| A$$
$$A \rightarrow \varepsilon$$
$$| A w$$

Assinale as afirmações verdadeiras.

Respostas selecionadas: Uma gramática equivalente a G, mas sem conflitos LR(0), é a seguinte:

☒ a. $S \rightarrow \varepsilon$
 $| S w$

☐ b. Se a produção `A $\rightarrow A w$ ` fosse alterada de forma a utilizar recursividade à direita, a gramática deixaria de ter conflitos shift/reduce.

☒ c. A gramática G apresenta 3 conflitos LR(0): 1 shift/reduce e 2 reduce/reduce.

Respostas corretas: Uma gramática equivalente a G, mas sem conflitos LR(0), é a seguinte:

☒ a. $S \rightarrow \varepsilon$
 $| S w$

☒ c. A gramática G apresenta 3 conflitos LR(0): 1 shift/reduce e 2 reduce/reduce.