UNIVERSIDADE FEDERAL DE SERGIPE
CAMPUS ITABAIANA
DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO
PROF: ANDRÉ LUIS MENESES SILVA
Aluno:_____ Data:04/12/2025

Primeira Avaliação de Linguagens Formais e Tradutores

1. Encontre os conjuntos primeiro e seguinte da gramática abaixo. Em seguida, construa a tabela do analisador sintático LL(1). (2,0)

Z → S $ *(1)*
S → A C A S *(2)* | A A C *(3)* | B *(4)*
A → C h C *(5)* | ε *(6)*
B → f B f *(7)* | p *(8)*
C → B C A *(9)* | ε *(10)*

P(Z) = P(S) - {ε} = {f, p, h, $}, todos gerados pela regra *(1)*
P(S) = P(A) U P(B) U P(C) U { ε } = {f, p, h, ε}, sendo {f, p, h, ε} da regra *(2)*, {f, p, h, ε} da regra *(3)* e {f, p} da regra *(4)*.
P(A) = P(C) U {ε} = {f, p, h, ε}, sendo {f, p, h} da regra *(5)* e {ε} da regra *(6)*
P(B) = {f, p}, sendo {f} da regra *(7)* e {p} da regra *(8)*.
P(C) = P(B) U {ε} = {f, p, ε}, sendo {f, p} da regra *(9)* e {ε} da regra *(10)*

S(Z) = { }
S(S) = {$}
S(A) = (P(S) U S(S) U P(C) U P(A) U S(C)) - {ε} = {f, p, h, $}
S(B) = (S(S) U {f} U P(C) U P(A) U S(C)) - {ε} = {f, p, h, $}
S(C) = (P(A) U P(S) U S(S) U {h} U S(A) U S(C)) - {ε} = {f, p, h, $}

	$	h	f	p
Z	1	1	1	1
S	2/3	2/3	2/3/4	2/3/4
A	6	5/6	5/6	5/6
B			7	8
C	10	10	9/10	9/10

2. Adotando o algoritmo visto em sala de aula, codifique um analisador sintático LL para a seguinte gramática livre de contexto. (2,0)

pg → f DEFfun | c DEFcmd | hyb DEFfun pg | jyb DEFcmd pg
DEFcmd → FUNC ID (parami) ; | ID = num ;
parami → ID ID | , ID ID
DEFfun → FUNC ID (parami) pg

https://github.com/andrelumesi/SINF0012-LFT/blob/main/2025.2/avaliacoes/unidade01/Quest02.java

3. Adotando a gramática da questão anterior, considere as seguintes regras: (2,0)

num	=	Começa com 0 ou 1 ou 2, seguido por quaisquer quantidades de 0 ou 1 e 2, finalizado com 3.
ID	=	Começa com o prefixo ID_ seguido por qualquer letra ou númeor.
		Os demais tokes são reconhecidos como eles ocorrem na gramática

a. Elabore código PLY que faça o reconhecimento do léxico dessa linguagem.

https://github.com/andrelumesi/SINF0012-LFT/blob/main/2025.2/avaliacoes/unidade01/quest03.py

4. Realize transformações nas gramáticas de acordo com o que se pede. (2,0)

a. precedência em ordem crescente e associatividade : 1. ADD direita 2.SUB esquerda 3.MULI direita 4. LESS esquerda. 5. !

exp → exp ADD exp | exp SUB exp | exp MULI exp | exp LESS exp | R exp | ID

exp → exp1 ADD exp | exp1
exp1 → exp1 SUB exp2 | exp2
exp2 → exp 3 MULI exp2 | exp3
exp3 → exp3 LESS exp4 | exp4
exp4 → R exp | ID

b. Eliminar recursividade à esquerda.

A → A p C | A d | r
C → d C | C C b | q

A → r A'
A' → p C A' | d A' | ε
C → d C C' | q C'
C' → C b C' | ε

c. Aplicar fatoração

bex → ID | ID ID ; | ID ID (typeids) | typeids ( ) bloco
typeids → TYPE ID , typeids | TYPE ID ; txt

bex → ID A | typeids ( ) bloco
A → ID B | ε
B → ; | (typeids)
typeids → TYPE ID C
C → , typeids | ; txt

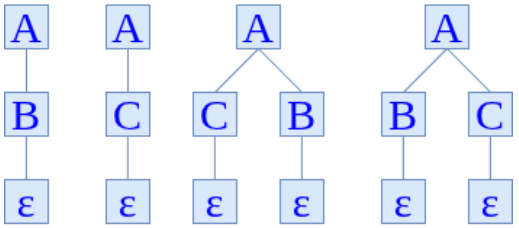
d. Justificar o motivo das seguintes gramáticas não serem LL(1).

I. A → + B | +C | BC
B → ID B | ID

II. A → B C | C B | B | C
B → a t s | p s C B | ε
C → d t s | t C | ε

I. A gramática I não é LL(1) pois Existem duas regras de A inicalizadas pelo mesmo token (+). Adicionalmente, duas de B também começam pelo mesmo token (ID).

II. A gramática II não é LL(1) pois é ambígua. Existem várias formas de se aceitar a palavra ε. A seguir, alguns exemplos de árvores de derivação que levam a ε.



5. Explique como a tabela preditiva é utilizada pelo analisador sintático LL(1) para realizar o reconhecimento ou não de possíveis entrada. Adote em sua explicação a gramática a seguir e o exemplo num + num - num.

E → T E′
E′ → + T E′
E′ →− T E′
E′ → num

Respostas aceitas:

1. A gramática não gera este padrão.
2. Para quem foi induzido ao erro, a questão foi desconsiderada e a pontuação redistribuida para as demais.