

”

E-fólio B | Folha de resolução para E-fólio



UNIDADE CURRICULAR: Linguagens e Computação

CÓDIGO: 21078

DOCENTE: Jorge Morais

A preencher pelo estudante

NOME: Pedro Pereira Santos

N.º DE ESTUDANTE: 2000809

CURSO: Licenciatura em Engenharia Informática

DATA DE ENTREGA: 29/12/2022

TRABALHO / RESOLUÇÃO:

Em anexo, na pasta .zip, envio as imagens originais utilizadas no relatório.

1.

$$\Sigma = \{g, l, o\}$$

$$g^z o^{3k+1} l^{2n} o^k \text{ onde } z > 0, k > 0, n > 0$$

Para escrever uma CFG G da forma $g^z o^{3k+1} l^{2n} o^k$ começamos por analisar como pode ocorrer tal sequência.

Como $z > 0, k > 0, n > 0$ sabemos que não existem sequências vazias de qualquer um dos terminais (com $z=0$ ou $k=0$ ou $n=0$). A sequência de g's aumenta conforme z pois não tem qualquer cálculo exponencial. Seguidamente vem a primeira sequência de o's onde podemos reparar que aumenta num fator de 3k, aumentando 3 o's por cada aumento unitário de k, sendo que o número de o's é um múltiplo de 3 mais 1 e, para a segunda sequência de o's, esta não tem, como para os g's, qualquer cálculo exponencial, aumentando conforme k. Para os l's, estes são sempre pares pois as sequências contêm sempre um número múltiplo de 2 de l's.

A tabela abaixo, mostra algumas palavras aceites pela gramática em questão.

z	k	n	g^z	o^{3k+1}	l^{2n}	o^k	palavra
1	1	1	g	oooo	ll	o	gooooollo
1	1	2	g	oooo	llll	o	goooollllo
1	2	1	g	oooooooo	ll	oo	gooooooooollo
2	1	1	gg	oooo	ll	o	ggooooollo
1	2	2	g	oooooooo	llll	oo	goooooooollllo
2	1	2	gg	oooo	llll	o	ggoooollllo
2	2	2	gg	oooooooo	llll	oo	ggoooooooollllo

Com isto, para a CFG G temos as produções:

$$S \rightarrow gS \mid gO$$

$$O \rightarrow oooOo \mid ooooLo$$

$$L \rightarrow ll \mid llL$$

Juntamos, então, estas produções num conjunto de produções denominado P.

Temos então as variáveis $V = \{S, O, L\}$ e os terminais $T = \{g, l, o\}$.

Logo, uma solução, segundo os dados obtidos, será:

$$G = \{V, T, P, S\}$$

As regras de produção S garantem que os termos de festejo comecem sempre por, pelo menos, um terminal 'g'. As regras de produção O garantem que a primeira sequência do terminal 'o' seja sempre um múltiplo de 3 mais 1, pelo requisito $3k+1$ e que a segunda sequência deste mesmo terminal aumente sempre em relação unitária a k. Por fim, a regra de produção L garante que o terminal 'l' seja apareça sempre me número par, no final da palavra.

Simulação da lista de festejos:

Simulação de Múltiplas Entrada(s)

Está no modo de simulação simples a entrada é testada a cada símbolo. Cada linha representará uma entrada a ser testada pelo simulador.

Simular

Limpar

gooooollo
ggoooooIIllo
ggoooooooooIIlloo
ggggooooooooooooIIlloo
ggggooooooooooooIIlloo
ggggooooooooooooIIlloo
ggggooooooooooooIIlloo

Análise e Resultados

Entrada	Resultado
g o o o o l l o	Sucesso!
g g o o o o l l l l o	Sucesso!
g g o o o o o o o l l l l o o	Sucesso!
g g g g o o o o o o o o o l l l l o o o	Insucesso!
g g g g o o o o o o o o o l l l l o o o	Insucesso!
g g g g o o o o o o o o o l l l l o o	Insucesso!
g g g o o o o o o o o l l l l l l o o	Sucesso!

Página 3 de 7

2.

$$\Sigma = \{g, o, l\}$$

$$P = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}, \{g, o, l\}, \{Z, g, o, l\}, \delta, q_0, Z, q_7)$$

Onde as transições são:

$$\delta(q_0, g, Z) = (\{q_0, gZ\})$$

$$\delta(q_0, g, g) = (\{q_0, g\})$$

$$\delta(q_0, o, g) = (\{q_1, \varepsilon\})$$

$$\delta(q_1, o, Z) = (\{q_2, oZ\})$$

$$\delta(q_2, o, o) = (\{q_3, o\})$$

$$\delta(q_3, o, o) = (\{q_4, o\})$$

$$\delta(q_4, l, o) = (\{q_5, lo\})$$

$$\delta(q_4, o, o) = (\{q_2, oo\})$$

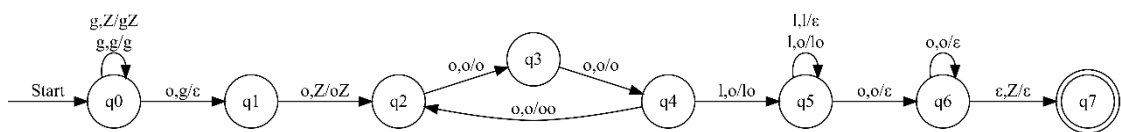
$$\delta(q_5, l, l) = (\{q_5, \varepsilon\})$$

$$\delta(q_5, l, o) = (\{q_5, lo\})$$

$$\delta(q_5, o, o) = (\{q_6, \varepsilon\})$$

$$\delta(q_6, o, o) = (\{q_6, \varepsilon\})$$

$$\delta(q_6, \varepsilon, Z) = (\{q_7, \varepsilon\})$$



O autômato desenhado é um PDA por pilha vazia ou “empty stack PDA”, iniciado no estado q_0 com o símbolo Z como símbolo inicial da stack. Ele inicia por adicionar um número indefinido de g 's à palavra, não sendo necessário

adicioná-los à pilha uma vez que a sua incógnita, z , não se associa a outro terminal. Uma vez que o número de g 's selecionado tenha sido adicionado passa-se para o estado q_1 ao ver o primeiro o , adicionando-o à stack como uma marca para poder comparar com os o 's após o terminal ' l '. Após isto, faz-se um ciclo entre 3 estados, q_2 , q_3 e q_4 para ir adicionando 3 o 's de cada vez uma vez que a condição $+1$ da função $3k+1$ foi feita na transição de q_0 para q_1 . Se se forem adicionar mais o 's após atingir o estado q_4 , ele retorna para o estado q_2 , adicionando mais um ' o ' à pilha para poder compará-lo com os o 's após os l 's. Isto acontece, pois, as funções têm a mesma incógnita ' k ', mas crescem de maneira diferente, onde a primeira sequência cresce em múltiplos de 3, ' $3k$ ' e a segunda cresce normalmente com o incremento de k pois só tem ' k '. A adição de um ' o ' na stack só acontece quando já se está no estado q_4 e de acrescenta mais o 's pois vai iniciar um novo ciclo de ' $3k$ '. No estado q_5 , faz-se o controlo para ter a certeza de que o número de l 's é par pois cresce de 2 em 2 segundo ' $2n$ '. Após verificar a condição dos l 's, e como o número de o 's foi controlado, adicionando somente k o 's na stack, o estado q_6 controla se o número de o 's da palavra coincide com $k-1$ (uma vez que um ' o ' já foi retirado na transição $q_5 \rightarrow q_6$), passando o autómato para o estado aceite após a palavra ter chegado ao fim e a pilha tiver apenas o Z inicial, eliminando-o nessa transição.

Tabela e Diagrama de Transições + Simulação de Múltiplas entradas

3.

Transitions Table

The lines represent the states, and the columns the possible inputs, the intercepts between lines and columns represent the state to which the automaton transits, during the simulation the corresponding intercepts are marked.

	ε	g	l	o
q0		Z/q0,gZ g/q0,g		g/q1,ε
q1				Z/q2,oZ
q2				o/q3,o
q3				o/q4,o
q4			o/q5,lo	o/q2,oo
q5			l/q5,ε o/q5,lo	o/q6,ε
q6	Z/q7,ε			o/q6,ε
q7				

Export: XLSX XLS CSV TXT

Transitions Diagram

Each node belongs to the set of states Q. The initial state is indicated by an arrow labeled 'Start' and each final state by a double node. During the simulation, the states and transitions are marked with different colors, in order to follow the simulation step-by-step.

Download Diagram Copy to Clipboard



Step-by-Step Input Simulation

Enter a sequence of symbols (without spaces) in the 'Input' field.
After starting the simulation by clicking on 'Simulate Input', click on 'Perform Step' to read each symbol in the sequence.
The table of transitions and the diagram of transitions will indicate the movements performed. You will also be able to see the stacks, entry positions and transitions for each existing possibility.

Insert Input

Simulate Input

Multiple Input Simulation

Enter a sequence of symbols (no spaces) on each line. Each line will represent an input to be tested by the simulator. The result will be displayed after clicking on 'Simulate'.

Sequence Generator

Simulate

Clean

gooooo
ggoooo
ggooooo
ggooooooooo
ggooooooooo
ggooooooooo
ggooooooooo
ggooooooooo

Input	Result
gooooo	Successful
ggoooo	Successful
ggooooo	Successful
ggooooooooo	Unsuccessful
ggooooooooo	Unsuccessful
ggooooooooo	Unsuccessful
ggooooooooo	Successful

Duas sequências que podem ser aceites pela linguagem construída são:

gggggggoooooooooooooooooooo

Transitions Table

The lines represent the states, and the columns the possible inputs, the intercepts between lines and columns represent the state to which the automaton transits, during the simulation the corresponding intercepts are marked.

	ε	g	l	o
q0		Z/q0,gZ g/q0,g		g/q1,ε
q1				Z/q2,oZ
q2				o/q3,o
q3				o/q4,o
q4			o/q5,lo	o/q2,oo
q5			l/q5,ε o/q5,lo	o/q6,ε
q6	Z/q7,ε			o/q6,ε
q7				

Export: XLSX XLS CSV TXT

Transitions Diagram

Each node belongs to the set of states Q. The initial state is indicated by an arrow labeled 'Start' and each final state by a double node. During the simulation, the states and transitions are marked with different colors, in order to follow the simulation step-by-step.

Download Diagram Copy to Clipboard



Step-by-Step Input Simulation

State	Successful!
-------	-------------

Input	gggggggoooooooooooooooooooo
-------	-----------------------------

In the 'Input' field your entry sequence is displayed.
Check the status of the stack(s), moves and transitions for each possible transition. By default, all possible transitions and states are displayed but only the first stack possibility is displayed, you can check the specific status of each stack and transition by selecting the desired transition from the drop down menu.

Possible Transitions (δ):

☒ All

☐ o,o/ε

☐ ε,Z/ε

1

gggggggooooollo

Multiple Input Simulation

Enter a sequence of symbols (no spaces) on each line. Each line will represent an input to be tested by the simulator. The result will be displayed after clicking on 'Simulate'.

Sequence Generator

Simulate Clean

gggggggoooooooooooooooooooo

Input	Result
gggggggoooooooooooooooooooo	Successful!

Simulação de Múltiplas Entrada(s)

Está no modo de simulação simples a entrada é testada a cada símbolo. Cada linha representará uma entrada a ser testada pelo simulador.

Simular Limpar

gggggggooooollo

Análise e Resultados

Entrada	Resultado
g g g g g o o o o l l o	Sucesso!
Regras de Produção	Aplicação
S → H0 S	S → H0 S
H0 → g	S → g S
S → H0 S	S → g H0 S
H0 → g	S → g g S
S → H0 S	S → g g H0 S
H0 → g	S → g g g S
S → H0 S	S → g g g H0 S
H0 → g	S → g g g g S
S → H0 S	S → g g g g H0 S
H0 → g	S → g g g g g S
S → H0 T	S → g g g g g H0 T
H0 → g	S → g g g g g g T
T → H2 H1	S → g g g g g g H2 H1
H2 → H6 U	S → g g g g g g H6 U H1
H6 → H7 H1	S → g g g g g g H7 H1 U H1
H7 → H8 H1	S → g g g g g g H8 H1 H1 U H1
H8 → H1 H1	S → g g g g g g H1 H1 H1 H1 U H1
H1 → o	S → g g g g g g o H1 H1 H1 U H1
H1 → o	S → g g g g g g o o H1 H1 U H1
H1 → o	S → g g g g g g o o o H1 U H1
H1 → o	S → g g g g g g o o o o U H1
U → H4 H4	S → g g g g g g o o o o H4 H4 H1
H4 → l	S → g g g g g g o o o o l H4 H1
H4 → l	S → g g g g g g o o o o l l H1
H1 → o	S → g g g g g g o o o o l l o