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

UNIDADE CURRICULAR: Linguagens e Computação

CÓDIGO: 21078

DOCENTE: Professor Jorge Morais

A preencher pelo estudante

NOME: Francisco José Pinto de Amaral

N.º DE ESTUDANTE: 1802876

CURSO: Licenciatura em Engenharia Informática

DATA DE ENTREGA: 01 - 01 - 2022

TRABALHO / RESOLUÇÃO:

Considere o alfabeto Σ = {a,b}.

1. Escreva uma gramática independente de contexto que reconheça todas as sequências da forma $a^{2n-1}b^{2k}a^{n}$, onde n e k são números inteiros positivos. Cotação: 1,5 valores.

De acordo com o enunciado sabemos que os valores exponenciais de n e k serão números inteiros positivos, desta forma não serão igualmente consideradas sequências vazias (n=0 e k=0). Substituindo os exponenciais temos como exemplo:

n	k	Expressão	Palavra
1	1	$a^1b^2a^1$	abba
2	1	$a^3b^2a^2$	aaabbaa
1	2	$a^1b^4a^1$	abbbba
2	2	$a^3b^4a^2$	aaabbbbaa
3	2	$a^5b^4a^3$	aaaaabbbbaaa
2	3	$a^3b^6a^2$	aaabbbbbbaa
3	3	$a^5b^6a^3$	aaaaabbbbbbaaa
…	…	…	…
5	4	$a^9b^8a^5$	aaaaaaaaabbbbbbbbaaaaa
…	…	…	…

De acordo com a tabela acima podemos verificar que a primeira sequência de a's é sempre ímpar (**abba**, **aaabbaa**, …), enquanto a sequência de b's é sempre par e múltiplo de 2 (**abba**, … , **abbbba**, … , **aaabbbbbbaa**, …), a ultima sequência de a's aumenta conforme o valor de n pois não tem qualquer cálculo do exponencial na formula facultada.

Deste modo tenho como solução as seguintes variáveis e inputs CFG:

S → aa**S**a | a**P**a

P → bb | bb**P**

Uma solução será:

G=({SP} , {a,b} , {S->aaSa | aPa , P->bb | bbP} , S)

Nesta definição G, que inicia com a variável S, garante que a palavra tem n a's no início e no final, assim como a P no meio que inicia com 2 b's mesmo na mínima expressão possível **abba** em que n=1 e k=1.

Para os testes de validação vou utilizar as 4 primeiras palavras/linhas da tabela:

abba

S -> aPa (S -> aPa)
-> abba (P -> bb)

abbbba

S -> aPa (S -> aPa)
-> abbPa (S -> bbP)
-> abbbba (P -> bb)

aaabbaa

S -> aaSa (S -> aaSa)
-> aaaPaa (S -> aPa)
-> aaabbaa (P -> bb)

aaabbbbaa

S -> aaSa (S -> aaSa)
-> aaaPaa (S -> aPa)
-> aaabbPaa (P -> bbP)
-> aaabbbbaa (P -> bb)

2. Construa um autómato de pilha que reconheça a mesma linguagem (sem recorrer à gramática da questão anterior). Cotação: 1,5 valores.

∑ = {a,b}

P = ({p,q,r,s,t}, {a,b}, {Z, a, b}, δ, p, Z, t)

Com as transições:

δ(p, a, Z) = ({p, aZ})

δ(p, a, a) = ({q, a})

δ(q, a, a) = ({p, aa})

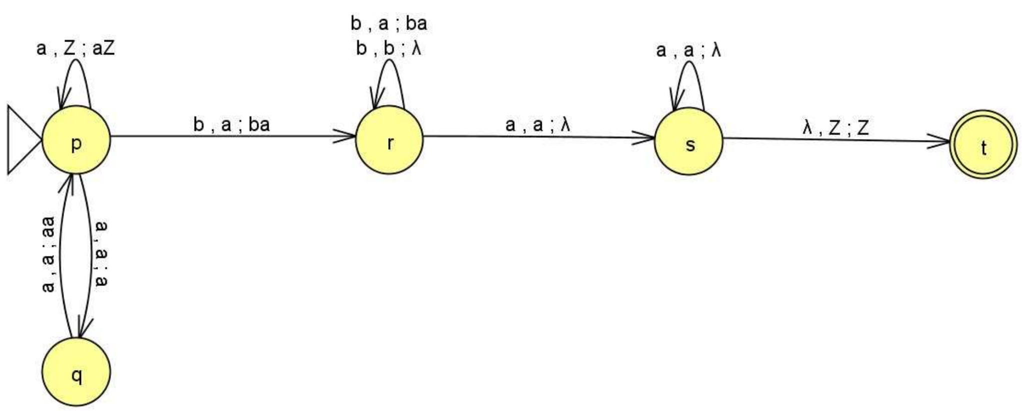
δ(p, b, a) = ({r, ba})

δ(r, b, b) = ({r, λ})

δ(r, a, a) = ({s, λ})

δ(s, a, a) = ({s, λ})

δ(s, λ, Z) = ({t, Z})



O autómato desenhado é um autómato de pilha vazia que inicia no estado p com o símbolo Z, inicialmente irá alternar entre os estados p e q caso seja necessário adicionar mais a's á pilha e igualmente nas transições entre os estados r → s e transição no estado s. Uma vez que os b's ficam entre os a's iniciais e finais, os b's são inseridos na transição entre os estados p → r e na transição no estado r. No final da palavra, se a pilha se encontrar vazia, incluindo o Z inicial, a palavra é reconhecida pelo autómato.

Foi utilizado o software JFLAP 7.1 para desenho do PDA e teste no mesmo software utilizando as palavras da tabela do exercício 1:

Input	Result
abba	Accept
aaabbaa	Accept
abbbba	Accept
aaabbbbaa	Accept
aaaaabbbbaaa	Accept
aaabbbbbbaa	Accept
aaaaabbbbbbaaa	Accept
aaaaaaaaabbbbbbbbaaaaa	Accept
bbaaaaaabbbb	Reject

NOTA: a última palavra **bbaaaaaabbbb** foi colocada propositadamente para mostrar que seria rejeitada.

3. Usando as duas respostas anteriores, uma diferente para cada sequência, mostre que abba e aaaaabbbbaaa pertence à respetiva linguagem. Cotação: 1 valor.

Utilizando a linguagem da questão 2 para a sequência **abba** temos:

OPERAÇÃO	INPUT	ESTADO INICIAL	TRANSIÇÃO	ESTADO FINAL
1ª			Z	p
2ª	a	p	(a,Z;aZ)	p
3ª	b	p	(b,a;ba)	r
4ª	b	r	(b,a;ba)	r
5ª	a	r	(a,a;λ)	s
6ª		s	(λ,Z;Z)	t

Utilizando a linguagem da questão 1 para a sequência **aaaaabbbbaaa** que é representado pela expressão $a^5b^4a^3$ em que n=3 e k=2 estrutura-se de acordo com as seguintes transições:

S-> aaSa (S -> aaSa)
-> aaaaSaa (S -> aaSa)
-> aaaaaPaaa (S -> aPa)
-> aaaaabbPaa (P -> bbP)
-> aaaaabbbbaa (P -> bb)

FIM