

”

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: 09 - 12 - 2021

TRABALHO / RESOLUÇÃO:

Considere o alfabeto $\Sigma = \{a, b, c\}$.

1. Construa e teste a expressão regular em notação UNIX para todas as sequências que começam em a, terminam em c, e todos os elementos aparecem na sequência pelo menos uma vez.

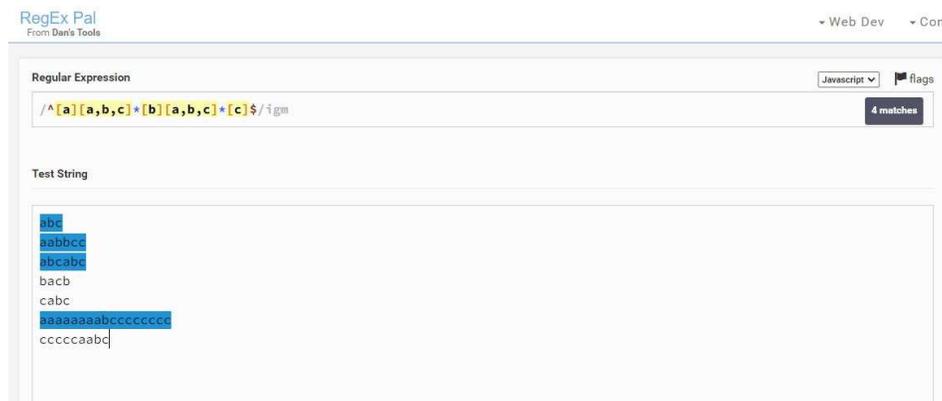
Para construção da expressão regular em notação UNIX tive em conta os seguintes pontos:

- A sequência deverá começar com um "a" podendo ter mais "a" dentro da sequência;
- A sequência deverá terminar com um "c" podendo ter mais "c" dentro da sequência;
- A sequência deverá ter pelo menos um "b" entre o "a" que marca o início da sequência e o "c" que marca o final da sequência;

Desta forma obtenho a seguinte expressão regular em notação UNIX:

$$[a][a,b,c]^*[b][a,b,c]^*[c]$$

Testando a expressão regular no RegExPal (<https://www.regexpal.com/>) com algumas strings por mim "inventadas" verifico que em todas as strings consigo ter sequências válidas conforme printouts infra:



Colocando o conteúdo do ficheiro "abc345.txt" partilhado pelo professor no fórum consigo obter 25 combinações.

2. Construa o NFA-ε correspondente, usando a construção de Thompson.

No seguimento da expressão regular anterior em notação UNIX, para desenho do NFA-ε correspondente dividi o mesmo em 5 blocos, sendo:

Bloco 1: [a]

Bloco 2: [a,b,c]*

Bloco 3: [b]

Bloco 4: [a,b,c]*

Bloco 5: [c]

Para uma melhor interpretação dos blocos coloquei mais o ε do que necessário, por exemplo poderia passar do q1 para o q3, no entanto no final o resultado é o mesmo.

Diagrama NFA-ε completo:

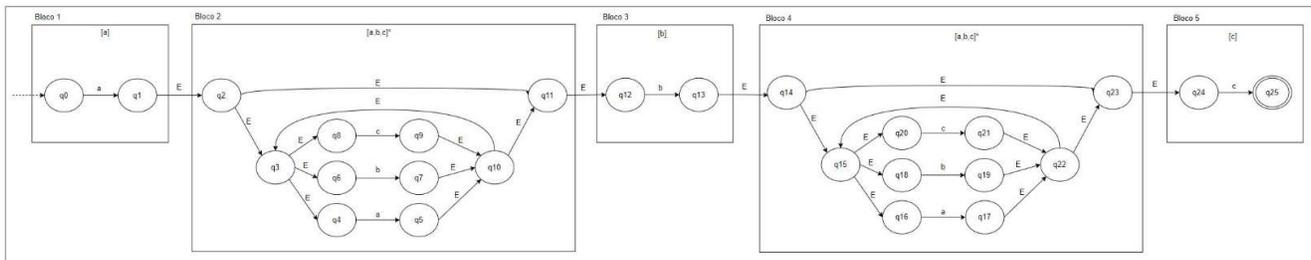
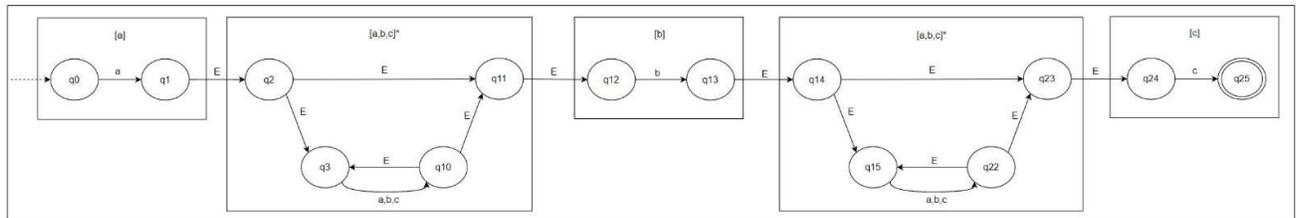


Diagrama NFA-ε simplificado:



Desenhos efetuados em draw.io (<https://app.diagrams.net/index.html>)

Referências bibliográficas:

Manual adotado: Seção 3.2.3 Converting Regular Expressions to Automata

<https://www.youtube.com/watch?v=DjH7K7MZRAw>

<https://www.youtube.com/watch?v=YGjEoND31YU>

3. Transforme-o num DFA, minimizando o número de estados.

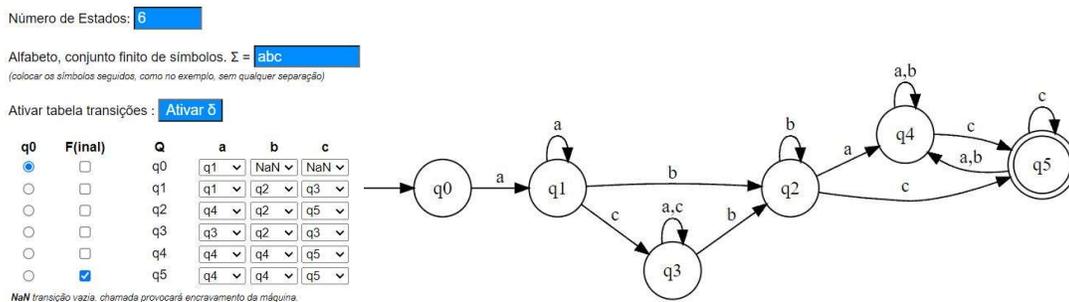
Para obter o DFA é necessário converter o NFA-ε calculando o ε-closure do estado inicial, q0, no entanto, uma vez que a nossa sequência tem que iniciar com um {a} o nosso estado inicial q0 não tem inicialmente ligação com mais nenhum, ou seja:

$$\varepsilon\text{-closure}(q_0) = \{q_0\}$$

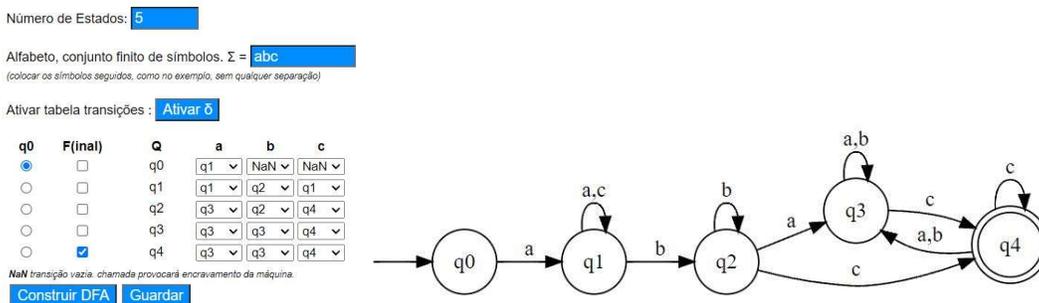
Nova Designação	Estado (q)	a	b	c
→q0 (A)	{0}	B	∅	∅
q1 (B)	{1, 2, 3, 4, 5, 6, 8, 10, 11, 12}	B	C	D
q2 (C)	{1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 22, 23, 24}	E	C	F
q3 (D)	{1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12}	D	C	D
q4 (E)	{1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 23, 24}	E	E	F
*q5 (F)	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25}	E	E	F

O símbolo * atrás do estado indica como sendo o estado final.

Desta forma obtém-se o respetivo DFA, ainda com 6 estados:



Tendo em conta que os estados q1 e q3 vão ambos para o estado q2 podem unir-se no mesmo estado considerando que tem obrigatoriamente que ter um b, ficando com 5 estados:



Mais uma vez temos dois estados (q2 e q3) com o mesmo estado de destino (q4) que conseguimos juntar em um só estado ficando com 4 estados:

Número de Estados: 4

Alfabeto, conjunto finito de símbolos. $\Sigma = abc$
(colocar os símbolos seguidos, como no exemplo, sem qualquer separação)

Ativar tabela transições: **Ativar δ**

q0	F(inal)	Q	a	b	c
<input checked="" type="radio"/>	<input type="checkbox"/>	q0	q1	NaN	NaN
<input type="radio"/>	<input type="checkbox"/>	q1	q1	q2	q1
<input type="radio"/>	<input type="checkbox"/>	q2	q2	q2	q3
<input type="radio"/>	<input checked="" type="checkbox"/>	q3	q2	q2	q3

NaN transição vazia, chamada provocará encerramento da máquina.

Construir DFA **Guardar**

Deste modo atingimos um limite mínimo de estados tendo como obrigatoriedade a inserção das 3 letras do nosso alfabeto $\Sigma = \{a, b, c\}$.

4. Usando a ferramenta UAbALL, teste todas as sequências possíveis do alfabeto de tamanho entre 3 e 5. Inclua no relatório uma imagem com o ecrã da aplicação para este DFA, bem como a tabela da simulação.

Usando a ferramenta UAbALL através do link [UAbALL](#) testando as sequências partilhadas pelo professor, ficheiro "abc345.txt", que utilizei na pergunta 1 deste eFólio, consigo obter as mesmas 25 combinações.

Printout da vista geral:

de um símbolo. É representado formalmente por uma 5-tupla $(Q, \Sigma, \delta, q_0, F)$, onde:

- Q é um conjunto finito de estados.
- Σ é um conjunto finito de símbolos, Alfabeto.
- δ é a função de transição, isto é, $\delta: Q \times \Sigma \rightarrow Q$.
- q_0 é o estado inicial, isto é, o estado do autómato antes de qualquer entrada ser processada, onde $q_0 \in Q$.
- F é um subconjunto de estados de Q (isto é, $F \subseteq Q$) chamado de estados de aceitação.

SIMULADOR

Número de Estados: 4 **Simular** **Limpar**

Alfabeto, conjunto finito de símbolos. $\Sigma = abc$
(colocar os símbolos seguidos, como no exemplo, sem qualquer separação)

Ativar tabela transições: **Ativar δ**

q0	F(inal)	Q	a	b	c
<input checked="" type="radio"/>	<input type="checkbox"/>	q0	q1	NaN	NaN
<input type="radio"/>	<input type="checkbox"/>	q1	q1	q2	q1
<input type="radio"/>	<input type="checkbox"/>	q2	q2	q2	q3
<input type="radio"/>	<input checked="" type="checkbox"/>	q3	q2	q2	q3

NaN transição vazia, chamada provocará encerramento da máquina.

Construir DFA **Guardar**

Simulação passo-a-passo

Insert input:

(colocar os símbolos seguidos sem qualquer separação)

Input consumido:

Projeto de Engenharia Informática 2020 de André Maciel Sousa

Printout do simulador onde coloco os parâmetros (já com os 4 estados):

Número de Estados: **4**

Alfabeto, conjunto finito de símbolos. $\Sigma =$ **abc**

(colocar os símbolos seguidos, como no exemplo, sem qualquer separação)

Ativar tabela transições : **Ativar δ**

q0	F(inal)	Q	a	b	c
<input checked="" type="radio"/>	<input type="checkbox"/>	q0	q1 ▾	NaN ▾	NaN ▾
<input type="radio"/>	<input type="checkbox"/>	q1	q1 ▾	q2 ▾	q1 ▾
<input type="radio"/>	<input type="checkbox"/>	q2	q2 ▾	q2 ▾	q3 ▾
<input type="radio"/>	<input checked="" type="checkbox"/>	q3	q2 ▾	q2 ▾	q3 ▾

NaN transição vazia. chamada provocará encravamento da máquina.

Construir DFA **Guardar**

Printout da tabela de simulação:
(para ser visualizado da seguinte forma:

[1] [3]
[2] [4]
...)

input	resultado	input consumido
aaa	Falha. Alcançou o estado não final q1	aaa[]
aab	Falha. Alcançou o estado não final q2	aab[]
aac	Falha. Alcançou o estado não final q3	aac[]
aba	Falha. Alcançou o estado não final q4	aba[]
abb	Falha. Alcançou o estado não final q2	abb[]
abc	Sucesso! Estado final alcançado q5	abc[]
aca	Falha. Alcançou o estado não final q3	aca[]
acb	Falha. Alcançou o estado não final q2	acb[]
acc	Falha. Alcançou o estado não final q3	acc[]
baa	Sem transição para o atual símbolo b no estado q0	[b]aa
bab	Sem transição para o atual símbolo b no estado q0	[b]ab
bac	Sem transição para o atual símbolo b no estado q0	[b]ac
bba	Sem transição para o atual símbolo b no estado q0	[b]ba
bbb	Sem transição para o atual símbolo b no estado q0	[b]bb
bbc	Sem transição para o atual símbolo b no estado q0	[b]bc
bca	Sem transição para o atual símbolo b no estado q0	[b]ca
bcb	Sem transição para o atual símbolo b no estado q0	[b]cb
bcc	Sem transição para o atual símbolo b no estado q0	[b]cc
caa	Sem transição para o atual símbolo c no estado q0	[c]aa
cab	Sem transição para o atual símbolo c no estado q0	[c]ab
cac	Sem transição para o atual símbolo c no estado q0	[c]ac
cba	Sem transição para o atual símbolo c no estado q0	[c]ba
cbb	Sem transição para o atual símbolo c no estado q0	[c]bb
cbc	Sem transição para o atual símbolo c no estado q0	[c]bc
cca	Sem transição para o atual símbolo c no estado q0	[c]ca
ccb	Sem transição para o atual símbolo c no estado q0	[c]cb
ccc	Sem transição para o atual símbolo c no estado q0	[c]cc
aaaa	Falha. Alcançou o estado não final q1	aaaa[]
aaab	Falha. Alcançou o estado não final q2	aaab[]
aaac	Falha. Alcançou o estado não final q3	aaac[]
aaaba	Falha. Alcançou o estado não final q4	aaaba[]
aabb	Falha. Alcançou o estado não final q2	aabb[]
aabc	Sucesso! Estado final alcançado q5	aabc[]
aaca	Falha. Alcançou o estado não final q3	aaca[]
aacb	Falha. Alcançou o estado não final q2	aacb[]
aacc	Falha. Alcançou o estado não final q3	aacc[]

abaa	Falha. Alcançou o estado não final q4	abaa[]
abab	Falha. Alcançou o estado não final q4	abab[]
abac	Sucesso! Estado final alcançado q5	abac[]
abba	Falha. Alcançou o estado não final q4	abba[]
abbb	Falha. Alcançou o estado não final q2	abbb[]
abbc	Sucesso! Estado final alcançado q5	abbc[]
abca	Falha. Alcançou o estado não final q4	abca[]
abcb	Falha. Alcançou o estado não final q4	abcb[]
abcc	Sucesso! Estado final alcançado q5	abcc[]
acaa	Falha. Alcançou o estado não final q3	acaa[]
acab	Falha. Alcançou o estado não final q2	acab[]
acac	Falha. Alcançou o estado não final q3	acac[]
acba	Falha. Alcançou o estado não final q4	acba[]
acbb	Falha. Alcançou o estado não final q2	acbb[]
acbc	Sucesso! Estado final alcançado q5	acbc[]
acca	Falha. Alcançou o estado não final q3	acca[]
accb	Falha. Alcançou o estado não final q2	accb[]
acc	Falha. Alcançou o estado não final q3	acc[]
baaa	Sem transição para o atual símbolo b no estado q0	[b]aaa
baab	Sem transição para o atual símbolo b no estado q0	[b]aab
baac	Sem transição para o atual símbolo b no estado q0	[b]aac
baba	Sem transição para o atual símbolo b no estado q0	[b]aba
babb	Sem transição para o atual símbolo b no estado q0	[b]abb
babc	Sem transição para o atual símbolo b no estado q0	[b]abc
baaca	Sem transição para o atual símbolo b no estado q0	[b]aaca
bacbc	Sem transição para o atual símbolo b no estado q0	[b]acbc
bacc	Sem transição para o atual símbolo b no estado q0	[b]acc
bbaa	Sem transição para o atual símbolo b no estado q0	[b]baa
bbab	Sem transição para o atual símbolo b no estado q0	[b]bab
bbac	Sem transição para o atual símbolo b no estado q0	[b]bac
bbba	Sem transição para o atual símbolo b no estado q0	[b]bba
bbbb	Sem transição para o atual símbolo b no estado q0	[b]bbb
bbbc	Sem transição para o atual símbolo b no estado q0	[b]bbc
bbca	Sem transição para o atual símbolo b no estado q0	[b]bca
bbcb	Sem transição para o atual símbolo b no estado q0	[b]bcb
bbcc	Sem transição para o atual símbolo b no estado q0	[b]bcc
bcaa	Sem transição para o atual símbolo b no estado q0	[b]caa
bcab	Sem transição para o atual símbolo b no estado q0	[b]cab

