

“

UNIDADE CURRICULAR: Linguagens e Computação

CÓDIGO: 21078

DOCENTE: Jorge Morais

NOME: Hélio Emanuel Soares de Sousa

N.º DE ESTUDANTE: 2000027

CURSO: Licenciatura em Engenharia Informática

DATA DE ENTREGA: 06 de Dezembro de 2020

TRABALHO / RESOLUÇÃO:

1. Construa e teste a expressão regular em notação UNIX para todas as sequências de vogais que contêm a subsequência "ai" ou "ui".

A resposta a esta pergunta é baseada nos conceitos dados na seção 3.3 de (Hopcroft, Motwani, & Ullmann, 2006), sempre que me referir a uma da secção é referente a este livro base da UC, e os testes de verificação foram executados em (RegExPal, 2020).

Primeiro farei com a notação do livro e depois farei a conversão para a notação UNIX. O alfabeto desta linguagem tem 5 símbolos possíveis (aeiou), todas as vogais. As subsequências “ai” e “ui” têm em comum a concatenação do “i”, logo podemos, simplificar para $(a + u)i$. Qualquer uma destas sequências pode ocorrer no início, meio ou no fim. Basta aparecer uma destas subsequências que a linguagem é aceite:

$(a+e+i+o+u)^* (a + u) i (a+e+i+o+u)^*$

Convertendo para a notação UNIX obtemos:

[aeiou]* (a|u)i [aeiou]*

ou

(a|e|i|o|u)* (u|a)i (a|e|i|o|u)*

Verifiquei as 156 combinações possíveis para palavras com 1, 2 e 3 símbolos, incluindo o ϵ , constata-se que esta expressão retorna as 22 palavras aceites, o que de fato está correto. A imagem retirada de (RegExPal, 2020) comprova o teste realizado.

Imagen - 1 Teste de verificação das palavras aceites

Desta forma pretendo assegurar que a expressão regular efetivamente realiza o pretendido.

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

No meu entendimento, o requerido nesta pergunta é a aplicação dos conceitos dados na seção 3.2.3 pois quando comparados com o conceito de construção de Thompson como definido em (WikiFoundation, 2020) são essencialmente os mesmos. Irei utilizar a notação dada no livro base. Assim sendo, $R = (a+e+i+o+u)^* (a + u) i (a+e+i+o+u)^*$

A estratégia de construção seguida foi a indicada no livro base, substituição direta dos passos indutivos apresentados na figura 3.17, isto é, aplicar os blocos base de construção a pequenos conjuntos de expressões regulares e, progressivamente englobar totalidade da expressão regular em causa, os passos seguidos foram:

1. Bloco 1 reunião: $(a+e+i+o+u)$
2. Acrescentar closure ao bloco 1: $(a+e+i+o+u)^*$
3. Cálculo auxiliar, criar o bloco 3 reunião: $(a + u)$
4. Concatenar o bloco 2 com o bloco 3: $(a+e+i+o+u)^* (a + u)$
5. Concatenar o bloco 4 com o símbolo i: $(a+e+i+o+u)^* (a + u) i$
6. Concatenar o bloco 5 com bloco 2: $(a+e+i+o+u)^* (a + u) i (a+e+i+o+u)^*$

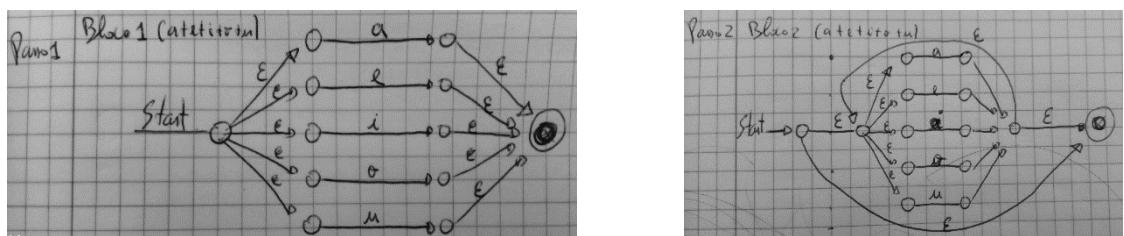


Imagen - 2 Primeiro bloco reunião: $(a+e+i+o+u)$ e acrescentar bloco closure ao bloco 1: $(a+e+i+o+u)^*$

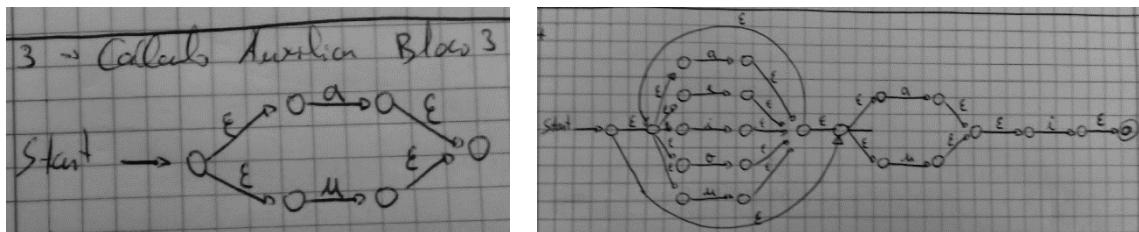


Imagen - 3 Cálculo auxiliar, criar o bloco reunião: $(a + u)$ e Concatenar o bloco 2 com o bloco reunião feito em 3 $(a+e+i+o+u)^* (a + u)$ e Concatenar o bloco 4 com o símbolo i: $(a+e+i+o+u)^* (a + u) i$

Ao resultado obtido por substituição direta designaremos por ϵ -NFA E:

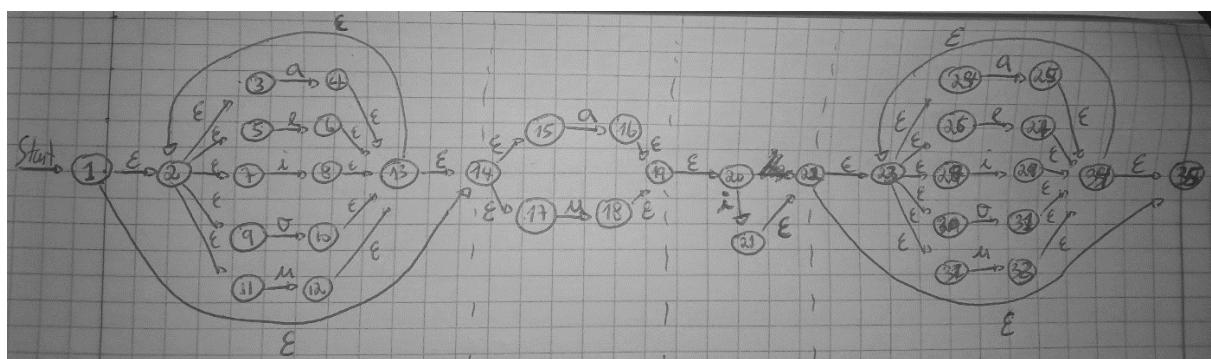


Imagen - 4 Concatenar bloco 5 com o bloco 2, ficando $(a+e+i+o+u)^* (a + u) i (a+e+i+o+u)^*$, obtendo-se desta forma o resultado final.

Tendo por base os conceitos dados em 2.5.1, podemos simplificar este ϵ -NFA E sabendo que $\Sigma = \{a, e, i, o, u\}$, mantendo o ciclo relativo ao conjunto $(a+e+i+o+u)$ simplificando de 35 para 12 estados, optei por este diagrama para na pergunta seguinte realizar as simplificações após a conversão para o DFA e poder aplicar o ECLOSE nesse processo.

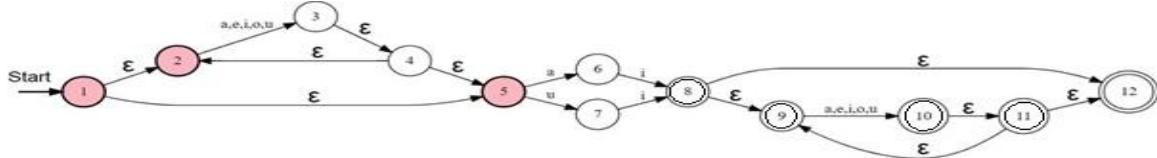


Imagen - 5 Diagrama de Transição

	ϵ	a	e	i	o	u
$\rightarrow 1$	{2,5}	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
2	\emptyset	{3}	{3}	{3}	{3}	{3}
3	{4}	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
4	{5}	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
5	\emptyset	{6}	\emptyset	\emptyset	\emptyset	{7}
6	\emptyset	\emptyset	\emptyset	{8}	\emptyset	\emptyset
7	\emptyset	\emptyset	\emptyset	{8}	\emptyset	\emptyset
8	{9,12}	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
9	\emptyset	{10}	{10}	{10}	{10}	{10}
10	{11}	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
11	{9,12}	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
12	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset

Tabela de Transição para o ϵ -NFA E

Para concluir defino formalmente o ϵ -NFA E:

$$\text{NFA E} = \{Q, \Sigma, \delta, q_0, F\}$$

$Q = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, 12 estados possíveis na atual forma deste ϵ -NFA.

$\Sigma = \{a, e, i, o, u\}$, alfabeto da linguagem.

δ = é função com argumentos $\Sigma \cup \epsilon$ definidos pela seu diagrama ou tabela de transição, definidos anteriormente

$q_0 = \{1\}$, estado inicial

$F = \{8, 9, 10, 11, 12\}$, conjunto de estados de aceitação da linguagem

$$\epsilon\text{-NFA E} = \{ \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}, \{a, e, i, o, u\}, \delta, \{1\}, \{8, 9, 10, 11, 12\} \}$$

E a linguagem deste ϵ -NFA E é definida da seguinte forma:

$$L(E) = \{w \mid \hat{\delta}(1, w) \cap \{8, 9, 10, 11, 12\} \neq \emptyset\}$$

w – é um conjunto de palavras contendo apenas vogais.

$\hat{\delta}(1, w)$ – é o conjunto de palavras w que contêm a subsequência “ai” ou “ui”, no inicio, meio ou fim.

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

Aplicando os conceitos dados em 2.5.4 calculei o ECLOSE(q) para cada um dos 12 estados do diagrama de transição do ϵ NFA E.

$$\text{ECLOSE}(1) = \{1,2,5\}$$

$$\text{ECLOSE}(5) = \{5\}$$

$$\text{ECLOSE}(9) = \{9\}$$

$$\text{ECLOSE}(2) = \{2\}$$

$$\text{ECLOSE}(6) = \{6\}$$

$$\text{ECLOSE}(10) = \{10\}$$

$$\text{ECLOSE}(3) = \{3\}$$

$$\text{ECLOSE}(7) = \{7\}$$

$$\text{ECLOSE}(11) = \{9,12\}$$

$$\text{ECLOSE}(4) = \{4,5\}$$

$$\text{ECLOSE}(8) = \{9,12\}$$

$$\text{ECLOSE}(12) = \{12\}$$

De seguida, aplico os conceitos dados em 2.5.5, a metodologia da AF2 e (Sintaha, 2020) para obter a tabela de transição do **DFA A** equivalente ao **ϵ NFA E**.

UAbALL	Estados	a	e	i	o	u
$\rightarrow q_0$	$\rightarrow [1,2,5]$	[2,3,4,5, 6]	[2,3,4,5]	[2,3,4,5]	[2,3,4,5]	[2,3,4,5, 7]
q_1	[2,3,4,5]	[2,3,4,5, 6]	[2,3,4,5]	[2,3,4,5]	[2,3,4,5]	[2,3,4,5, 7]
q_2	[2,3,4,5,6]	[2,3,4,5, 6]	[2,3,4,5]	[2,3,4,5, 8,9,12]	[2,3,4,5]	[2,3,4,5, 7]
q_3	[2,3,4,5,7]	[2,3,4,5, 6]	[2,3,4,5]	[2,3,4,5, 8,9,12]	[2,3,4,5]	[2,3,4,5, 7]
$*q_4$	[2,3,4,5, 8,9,12]	[2,3,4,5, 6, 9,10,11,12]	[2,3,4,5, 9,10,11,12]	[2,3,4,5, 9,10,11,12]	[2,3,4,5, 9,10,11,12]	[2,3,4,5, 7, 9,10,11,12]
$*q_5$	[2,3,4,5, 9,10,11,12]	[2,3,4,5, 6, 9,10,11,12]	[2,3,4,5, 9,10,11,12]	[2,3,4,5, 9,10,11,12]	[2,3,4,5, 9,10,11,12]	[2,3,4,5, 7, 9,10,11,12]
$*q_6$	[2,3,4,5, 6, 9,10,11,12]	[2,3,4,5, 6, 9,10,11,12]	[2,3,4,5, 9,10,11,12]	[2,3,4,5, 8 9,10,11,12]	[2,3,4,5, 9,10,11,12]	[2,3,4,5, 7, 9,10,11,12]
$*q_7$	[2,3,4,5, 7, 9,10,11,12]	[2,3,4,5, 6, 9,10,11,12]	[2,3,4,5, 9,10,11,12]	[2,3,4,5, 8 9,10,11,12]	[2,3,4,5, 9,10,11,12]	[2,3,4,5, 7, 9,10,11,12]
$*q_8$	[2,3,4,5, 8 9,10,11,12]	[2,3,4,5, 6, 9,10,11,12]	[2,3,4,5, 9,10,11,12]	[2,3,4,5, 9,10,11,12]	[2,3,4,5, 9,10,11,12]	[2,3,4,5, 7, 9,10,11,12]

DFA A obtido:

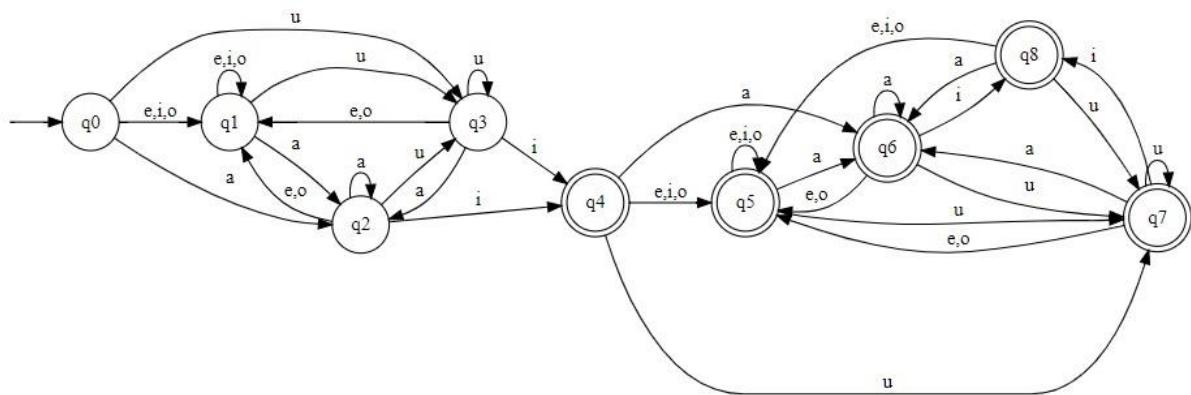


Imagen - 6 – DFA A (9 estados)

Obteve-se 9 estados, mas é possível simplificar. A começar pela quantidade de estados aceitáveis, podemos eliminar 4 estados, pois após atingir o estado de aceitação o DFA permanecerá sempre em estado de aceitação, pois a subsequência já foi encontrada.

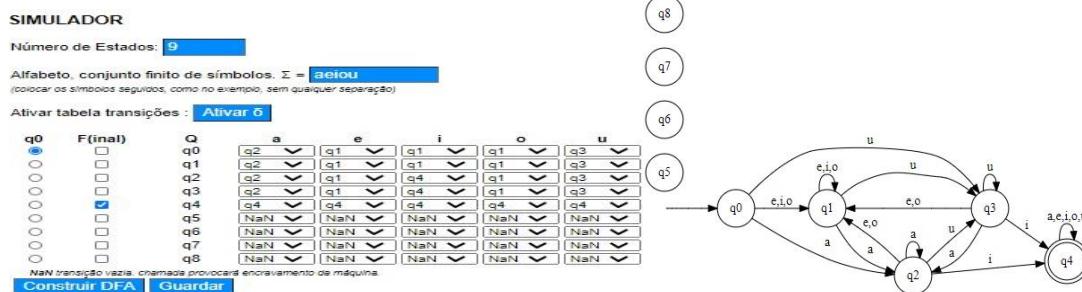


Imagen - 7 DFA A (5 estados)

Agora vamos analisar o estado inicial e o q_1 , essencialmente são o mesmo, é uma transição redundante, pode ficar para o próprio estado inicial, com todas as restantes transições a serem reajustadas.

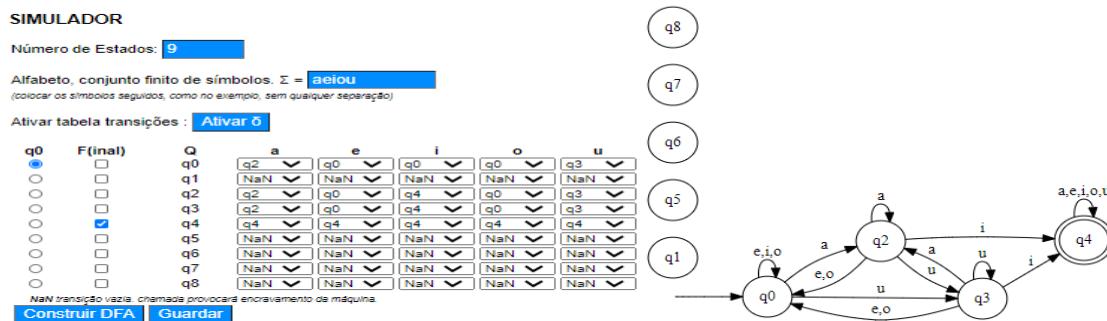


Imagen - 8 DFA A 4 estados

É possível simplificar até 3 estados, agrupando as transições de e para o q_2 e q_3 em apenas um único estado.

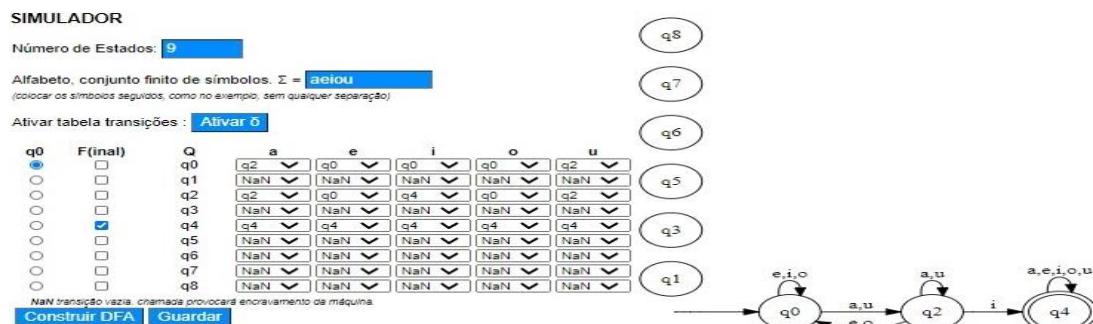


Imagen - 9 DFA A (3 estados)

Mantive as nomeações inalteradas para melhor se perceber os passos de cada simplificação realizada, agora no final para obter alguma coerência no DFA apresento o mesmo renomeado e sem os estados redundantes:

SIMULADOR

Número de Estados:

Alfabeto, conjunto finito de símbolos. $\Sigma = \text{aeiou}$

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

Ativar tabela transições :

q_0	$F(\text{final})$	Q	a	e	i	o	u
<input checked="" type="radio"/>	<input type="checkbox"/>	q_0	$q_1 \downarrow$	$q_0 \downarrow$	$q_0 \downarrow$	$q_0 \downarrow$	$q_1 \downarrow$
<input type="radio"/>	<input type="checkbox"/>	q_1	$q_1 \downarrow$	$q_0 \downarrow$	$q_2 \downarrow$	$q_0 \downarrow$	$q_1 \downarrow$
<input type="radio"/>	<input checked="" type="checkbox"/>	q_2	$q_2 \downarrow$				

NaN transição vazia, chamada provocará encravamento da máquina.

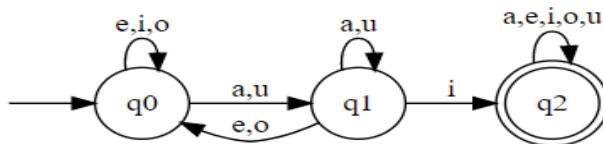


Imagen - 10 DFA A (3 estados) renomeado

Para concluir defino formalmente o DFA A equivalente ao ϵ -NFA E :

$$\text{DFA A} = \{Q, \Sigma, \delta, q_0, F\}$$

$Q = \{q_0, q_1, q_2\}$, 3 estados possíveis para o DFA A

$\Sigma = \{a, e, i, o, u\}$, alfabeto da linguagem.

δ = É a função de transição com argumentos contidos em Σ definidos pelo seu diagrama ou tabela de transição, ver imagem anterior.

$q_0 = \{q_0\}$, estado inicial

$F = \{q_2\}$, conjunto de estados de aceitação da linguagem

$$\text{DFA A} = \{ \{q_0, q_1, q_2\}, \{a, e, i, o, u\}, \delta, \{q_0\}, \{q_2\} \}$$

E a linguagem deste DFA A é definida da seguinte forma:

$$L(A) = \{w \mid \hat{\delta}(q_0, w) \cap \{q_2\} \neq \emptyset\}$$

w – é um conjunto de palavras contendo apenas vogais.

$\hat{\delta}(1, w)$ – é o conjunto de palavras w que contêm a subsequência “ai” ou “ui”, no inicio, meio ou fim.

4. Usando a ferramente UAbALL, teste todas as sequências do alfabeto de tamanho menor ou igual a 3. Inclua no relatório uma imagem com o ecrã da aplicação para este DFA, bem como a tabela da simulação

Em baixo, os testes realizados a todas as 156 palavras verificando-se o retorno das 22 palavras aceitáveis nesta linguagem para o DFA A, o retorno é o mesmo para as 4 versões (9,5,4 e 3 estados). Fui testando à medida que ia realizando a simplificação.

BIBLIOGRAFIA

Hopcroft, J., Motwani, R., & Ullmann, J. (2006). *Introduction to Automata Theory, Languages and Computation* (3 ed.). Boston: Pearson Education, Inc.

NesoAcademy. (29 de 11 de 2020). *Youtube*. Obtido de Conversion of Epsilon NFA to NFA: <https://www.youtube.com/watch?v=WSGcmaHNBFM>

RegExPal. (27 de 11 de 2020). *RegExPal*. Obtido de RegExPal from Dan's Tols: <https://www.regexpal.com/>

Sintaha, M. (29 de 11 de 2020). *Youtube*. Obtido de E-Closure : E-NFA to DFA Conversion: <https://www.youtube.com/watch?v=oEraHUCwFVU>

WikiFoundation. (27 de 11 de 2020). *Thompson's construction*. Obtido de Wikipedia: https://en.wikipedia.org/wiki/Thompson%27s_construction