

”

**E-fólio A** | 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:** 05/12/2022

## TRABALHO / RESOLUÇÃO:

### 1.

Considerando o alfabeto  $\Sigma = \{0,1\}$ , para construir a expressão regular para todas as sequências que não têm dois 1's seguidos, consideremos os seguintes pontos:

- Como a palavra vazia não contém dois 1's seguidos (11) esta é aceite.
- Para garantir que não existem dois 1's seguidos, temos a expressão **(10+0)\***, que nunca deixa que dois 1's estejam seguidos, escolhendo entre a sequência 10 ou 0, zero ou mais vezes, obtendo, por exemplo, 101010 ou 010, mas nunca, por exemplo, 10110.
- No fim, a expressão poderá ou não ter um 1, ou seja **(1+ε)**

Obtemos, assim a expressão regular: **(10+0)\*(1+ε)**

Esta expressão tem o equivalente, em UNIX, de: **(10|0)\*1?**

No entanto, também podíamos ter usado a expressão: **1?(01|0)\***

que é semelhante à anterior.

Para os testes, utilizei sequências de tamanho inferior a 4, como pedido na questão 4, e mais algumas criadas por mim, incluindo a palavra vazia, totalizando 21 sequências, e testei no regex101, utilizando ^ e \$, no início e fim da expressão, respetivamente, para melhor avaliar as palavras por linha, pela funcionalidade de "multiline" da ferramenta. Obtive 14 sequências aceites, sendo estas as previstas a passarem o teste.

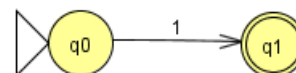
Em baixo mostro a imagem do mesmo.



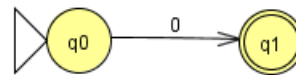
**2.**

Utilizando o algoritmo pedido em conjunto com as regras definidas no manual adotado para a construção do NFA-ε a partir da expressão regular, divide-se a execução a várias partes mostrando as várias transições:

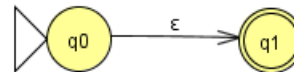
- Representação do símbolo "1"



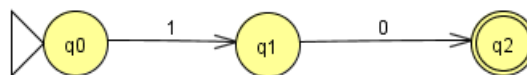
- Representação do símbolo "0"



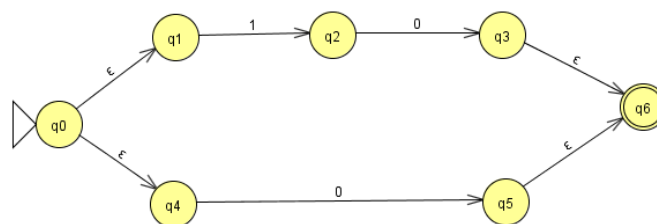
- Representação do símbolo "ε"



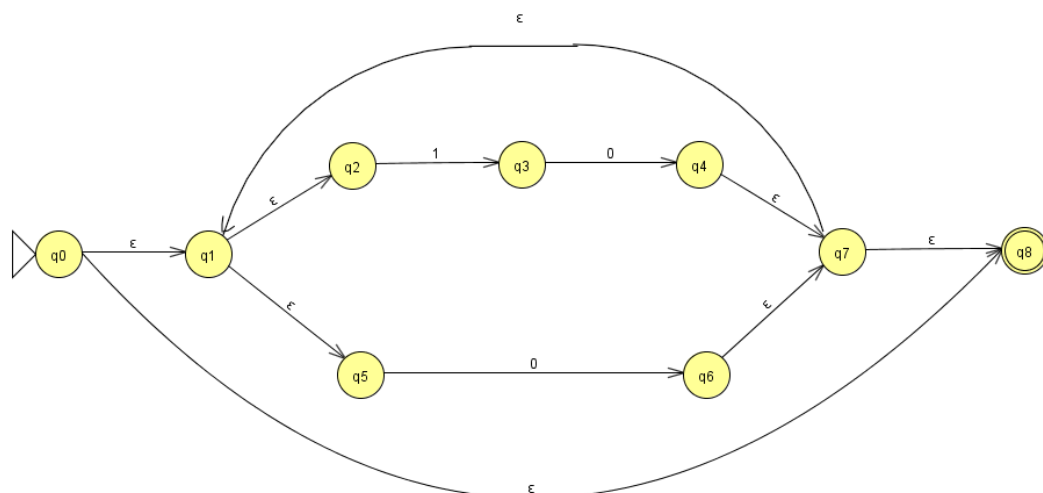
- Representação da concatenação de "1" com "0": **10**



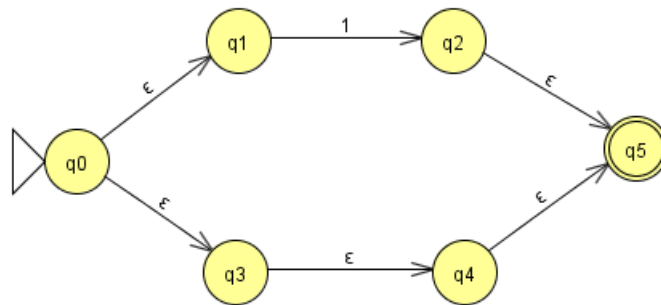
- Representação da união de "10" com "0": **10+0**



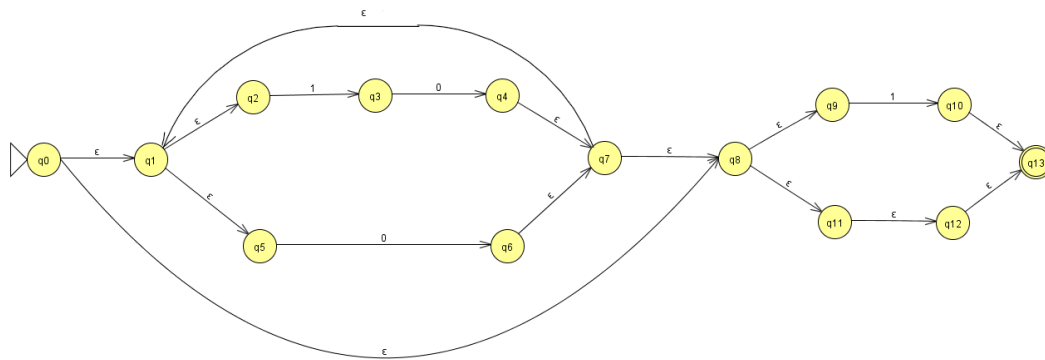
- Representação do fecho de Kleene de "(10+0)": **(10+0)\***



- Representação da união de "1" com "ε": **1+ε**



- Representação do ε-NFA da expressão regular **(1+0)\*(1|ε)**:



Desenhos efetuados na aplicação JFLAP (<https://www.jflap.org/>).

### 3.

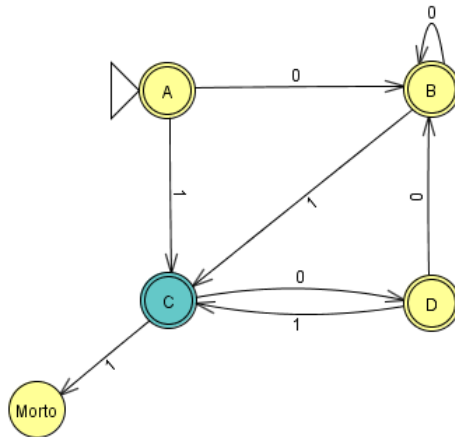
Começamos por fazer o fecho- $\epsilon(q_0)$ .

Temos então  $\text{fecho-}\epsilon(q_0) = \{q_0, q_1, q_2, q_5, q_8, q_9, q_{11}, q_{12}, q_{13}\} = A$

Logo, temos:

$\delta$	0	1
$\rightarrow *A$	$\text{fecho-}\epsilon(q_6) = \{q_1, q_2, q_5, q_6, q_7, q_8, q_9, q_{11}, q_{12}, q_{13}\} = B$	$\text{fecho-}\epsilon(q_3 \cup q_{10}) = \{q_3, q_{10}, q_{13}\} = C$
*B	$\text{fecho-}\epsilon(q_6) = B$	$\text{fecho-}\epsilon(q_3 \cup q_{10}) = C$
*C	$\text{fecho-}\epsilon(q_4) = \{q_1, q_2, q_4, q_5, q_7, q_8, q_9, q_{11}, q_{12}, q_{13}\} = B$	$\emptyset$
*D	$\text{fecho-}\epsilon(q_6) = B$	$\text{fecho-}\epsilon(q_3 \cup q_{10}) = C$

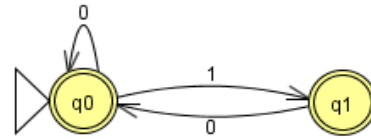
Obtemos, então, o seguinte DFA com 4 estados, mais o estado "Morto":



Para minimizar, temos de separar primeiros os estados finais de não finais, mas todos são finais. Podemos então ver que os estados A, B e D são equivalentes, ficando assim com dois estados  $q_0 = \{A, B, D\}$  e  $q_1 = \{C\}$ . Podemos, também, ver que o estado C têm a transição para o estado "Morto" (ou vazio " $\emptyset$ "), podendo, segundo as regras de transformação de NFA em DFA, retirar este estado "Morto" ou chamado de TRAP, pois não leva a nenhum estado de aceitação.

Ficamos, então, com o seguinte DFA Minimizado, com a sua tabela de transições:

	0	1
q0	q0	q1
q1	q0	$\emptyset$



Desenhos efetuados na aplicação JFLAP (<https://www.jflap.org/>).

#### 4.

Teste de todas as sequências possíveis do alfabeto de tamanho inferior a 4:

```

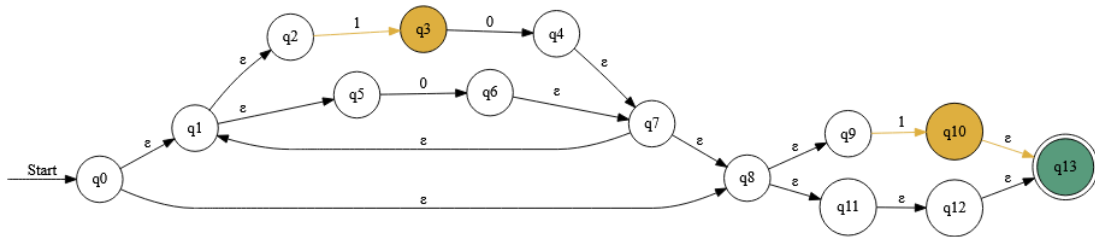
0
1
00
01
10
11
000
001
010
011
  
```

Entrada	Resultado
0	Sucesso!
1	Sucesso!
00	Sucesso!
01	Sucesso!
10	Sucesso!
11	Insucesso!
000	Sucesso!
001	Sucesso!
010	Sucesso!
011	Insucesso!
100	Sucesso!
101	Sucesso!
110	Insucesso!
111	Insucesso!
$\epsilon$	Sucesso!

## Simulação para o NFA-ε:

Simulação de Entrada Passo-a-Passo

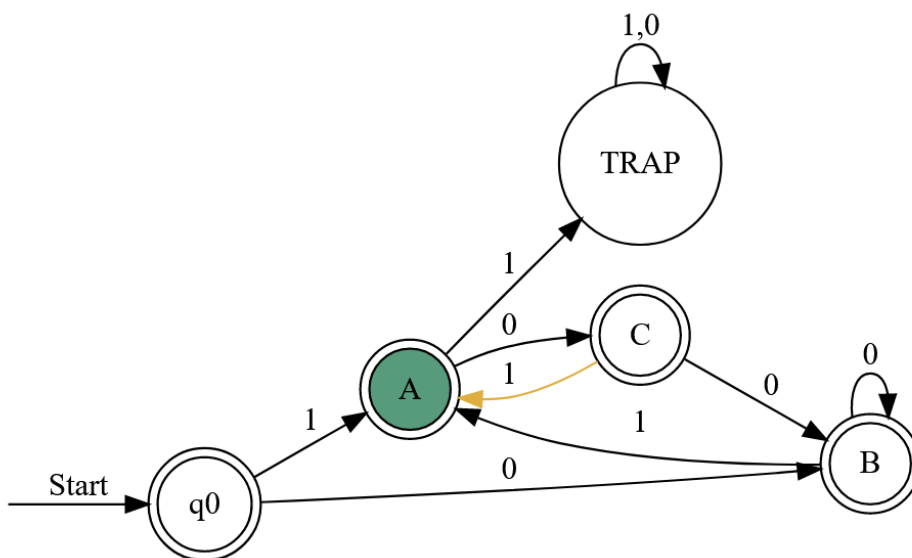
Estado	Sucesso!
Entrada	101



## Simulação para o DFA:

Simulação de Entrada Passo-a-Passo

Estado	Sucesso!
Entrada	101





## Simulação para o DFA minimizado:

Simulação de Entrada Passo-a-Passo

Estado	Sucesso!
Entrada	101

