

# Teoria da Computação

2022/1 - Trabalho da Disciplina

## Enunciado

Implemente, em C ou C++, um simulador de autômatos não determinísticos com transições  $\varepsilon$  ( $\varepsilon$ -NFAs). Seu programa deve ler do usuário a descrição de um  $\varepsilon$ -NFA seguida de uma sequência de palavras, e determinar, para cada palavra, se ela pertence ou não à linguagem que o  $\varepsilon$ -NFA reconhece. Em caso positivo, seu programa também deve imprimir a sequência de estados pelos quais a máquina deve passar para aceitar a palavra.

## Especificação da entrada e saída

A primeira linha da entrada contém um inteiro  $A$  indicando o tamanho do alfabeto  $\Sigma$ , seguido de uma string de  $A$  caracteres indicando, cada um, uma letra de  $\Sigma$ .

A segunda linha contém um inteiro  $Q$  indicando o número de estados. Considere que os estados são numerados de 0 a  $Q - 1$ , e que o estado inicial é sempre o estado 0.

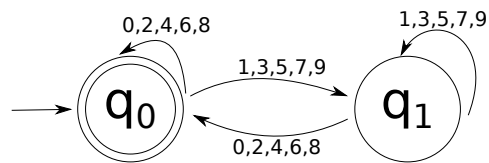
A próxima linha contém um inteiro  $F$  indicando o número de estados finais, seguido de  $F$  inteiros indicando os estados finais em si.

As próximas linhas descrevem a função de transição  $\delta$ . Cada linha é dada no formato **X str Y**, onde  $X$  e  $Y$  são estados e  $str$  é uma string. Cada letra  $l$  da string  $str$  indica uma transição do estado  $X$  para o estado  $Y$  com a letra  $l$  (isto é,  $Y \in \delta(X, l)$ ). Transições vazias ( $\varepsilon$ ) são representadas na string pela letra **&**. A descrição da função de transição termina com a linha **-1 \* -1**.

Por fim, a entrada contém uma sequência de palavras, uma por linha. Para cada palavra  $w$  dada na entrada, seu programa deve imprimir uma linha contendo **w: nao** se o  $\varepsilon$ -NFA rejeita  $w$ , ou **w: sim** se o  $\varepsilon$ -NFA aceita  $w$ . Em caso positivo, seu programa também deve imprimir, na ordem em que os estados são visitados pelo autômato, linhas no formato **X -> 1 -> Y**, indicando que a máquina, no estado  $X$  processando a letra  $l$ , muda para o estado  $Y$ .

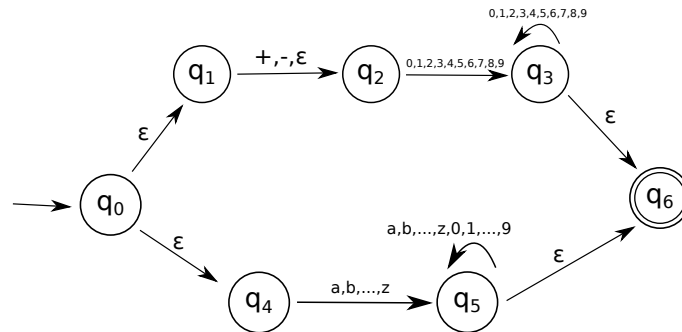
Imprima uma linha em branco após cada palavra. O final da entrada é indicado por uma linha contendo apenas **\***.

Como exemplo, considere  $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ . O seguinte autômato de exemplo (que reconhece números pares) é descrito pelo exemplo de entrada abaixo:



Exemplo de entrada	Exemplo de saída
10 0123456789	42: sim
2	0 -> 4 -> 0
1 0	0 -> 2 -> 0
0 02468 0	
0 13579 1	3685: nao
1 13579 1	
1 02468 0	194: sim
-1 * -1	0 -> 1 -> 1
42	1 -> 9 -> 1
3685	1 -> 4 -> 0
194	
0	0: sim
33	0 -> 0 -> 0
*	
	33: nao

Como outro exemplo, considere  $\Sigma = \{+, -, 0, 1, \dots, 9, a, b, \dots, z\}$ . O seguinte autômato de exemplo (referente à solução do exercício 10(c) da lista 1) é descrito pelo exemplo de entrada abaixo:



### Exemplo de entrada

```

38 +-0123456789qwertyuioplkjhgfdsazxcvbnm
7
1 6
0 & 1
1 +-& 2
2 0123456789 3
3 0123456789 3
3 & 6
0 & 4
4 abcdefghijklmnopqrstuvwxyz 5
5 abcdefghijklmnopqrstuvwxyz0123456789 5
5 & 6
-1 * -1
42
-328
+9
0
+0
-0
--256
+
2-83+1
i
evair
x1y2
42
1x2y
*
```

### Exemplo de saída

```
42: sim
0 -> & -> 1
1 -> & -> 2
2 -> 4 -> 3
3 -> 2 -> 3
3 -> & -> 6

--256: nao

+: nao

2-83+1: nao

-328: sim
0 -> & -> 1
1 -> - -> 2
2 -> 3 -> 3
3 -> 2 -> 3
3 -> 8 -> 3
3 -> & -> 6

i: sim
0 -> & -> 4
4 -> i -> 5
5 -> & -> 6

evair: sim
0 -> & -> 4
4 -> e -> 5
5 -> v -> 5
5 -> a -> 5
5 -> i -> 5
5 -> r -> 5
5 -> & -> 6

+9: sim
0 -> & -> 1
1 -> + -> 2
2 -> 9 -> 3
3 -> & -> 6

x1y2: sim
0 -> & -> 4
4 -> x -> 5
5 -> 1 -> 5
5 -> y -> 5
5 -> 2 -> 5
5 -> & -> 6

0: sim
0 -> & -> 1
1 -> & -> 2
2 -> 0 -> 3
3 -> & -> 6

42: sim
0 -> & -> 1
1 -> & -> 2
2 -> 4 -> 3
3 -> 2 -> 3
3 -> & -> 6

1x2y: nao

+0: sim
0 -> & -> 1
1 -> + -> 2
2 -> 0 -> 3
3 -> & -> 6

-0: sim
0 -> & -> 1
1 -> - -> 2
2 -> 0 -> 3
3 -> & -> 6
```

O seu programa será testado com outras entradas além das dadas como exemplo. Você pode assumir que o  $\varepsilon$ -NFA terá no máximo 100 estados, que o alfabeto terá no máximo 100 letras, e que a entrada será sempre válida (em particular, que todas as strings dadas na entrada são formadas apenas por letras do alfabeto dado, e que  $\&$  e  $*$  nunca estarão no alfabeto dado).

## Orientações

- O trabalho pode ser feito por equipes de *até* 2 (dois) estudantes;
- Submeta, via *Moodle*, um pacote (zip ou tar.gz) contendo todos os arquivos necessários para compilar e executar seu programa, além de um arquivo de texto (txt) onde conste:
  - O nome de todos os integrantes da equipe;
  - Toda informação que a equipe julgar relevante para a correção (como *bugs* conhecidos, detalhes de implementação, escolhas de projeto, etc.)
- Comente adequadamente seus códigos para facilitar a correção.
- Atenção: a correção será parcialmente automatizada, e a saída do programa será testada com outras entradas além das fornecidas como exemplo. *Siga **fielmente** o formato de saída dado nos exemplos*, sob pena de grande redução da nota;
- Certifique-se que seu programa compila e funciona antes de submetê-lo;
- O trabalho deve ser entregue até **8 de Maio de 2022, 23:59**, apenas via *Moodle*. Trabalhos entregues por outros meios ou fora do prazo não serão aceitos. É suficiente que o trabalho seja submetido por apenas um estudante da equipe;
- Trabalhos detectados como cópia, plágio ou comprados receberão **todos** a nota 0 (**ZERO**) e estarão sujeitos a abertura de Processo Administrativo Disciplinar Discente.