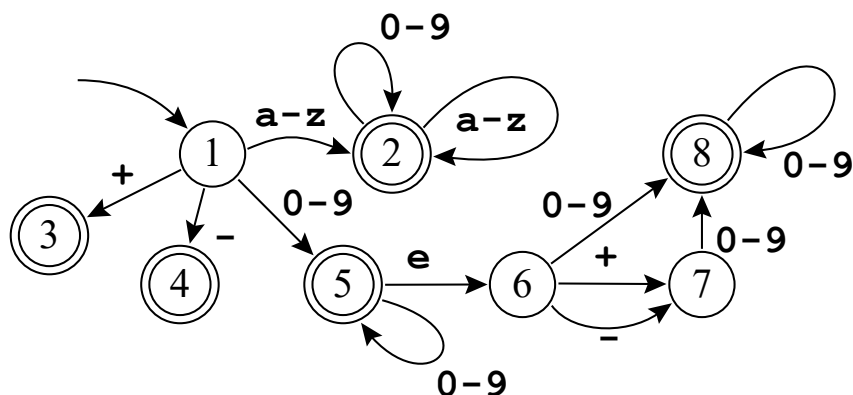


## 5COP093 - Lista de Exercícios 05

1.  **Exercício Prático:** Escreva um programa que implementa o seguinte autômato:



O programa deve ler a sua entrada direto do teclado e imprimir um *token* por linha e chegar ao fim ao encontrar o símbolo de fim de arquivo. Para caracteres que não façam parte do alfabeto, o programa deve imprimir a mensagem ERRO. Quebras de linha devem ser ignoradas. Exemplo:



**Entrada:**



```
+-abcabc@ abc
fgh
```


**Saída:**

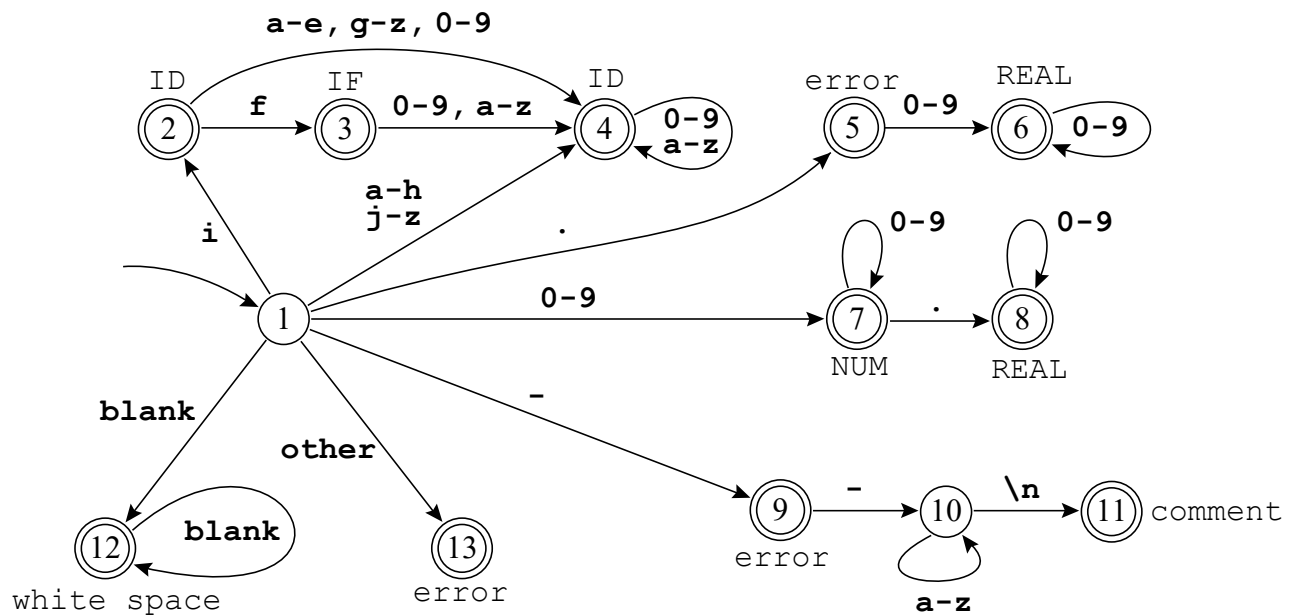
```
+
-
abcabc
ERRO
ERRO
abc
fgh
```

O símbolo de fim de arquivo pode ser emulado no teclado da seguinte forma:

Windows:  + 

Linux:  + 

2.  **Exercício Prático:** Implemente um analisador Léxico para o autômato a seguir:



O programa deve ler a sua entrada direto do teclado e imprimir um *token* por linha e chegar ao fim ao encontrar o símbolo de fim de arquivo. Quebras de linha que não estejam associadas ao *token comment* devem ser ignoradas. Exemplo:







**Entrada:**


```
chuchu 666abobrinha@--blabla
0.0
```

**Saída:**

```
chuchu ID
white space
666 NUM
abobrinha ID
@ error
--blabla comment
0.0 REAL
```

O símbolo de fim de arquivo pode ser emulado no teclado da seguinte forma:

Windows:   +   
Linux:   + 

3.  **Exercício Teórico:** Construa um autômato finito determinístico que aceita as cadeias geradas pelas expressões regulares a seguir, as quais estão apresentadas em ordem de prioridade (da maior para a menor):

$(aba)^+$	(action 1)
$a(b^*)a$	(action 2)
$a b$	(action 3)

Com o autômato construído, monte a tabela de análise léxica e mostre todos os passos necessários para reconhecer os *tokens* na cadeia `abaabbaba` e diga ao final do reconhecimento como cada *token* é classificado, isto é: (action 1), (action 2) ou (action 3).