

Etude de cas :

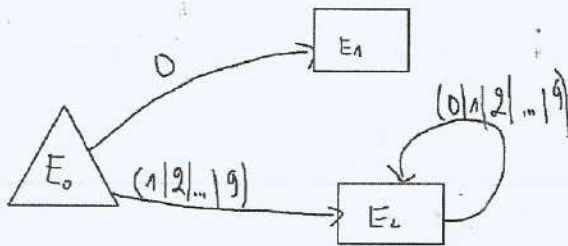
Nous allons modéliser une expression régulière qui dénote tous les entiers naturels,

Pour cela il va falloir

1. Proposer une Expression régulière
2. Traduire l'ER en un automate
3. Déterminer l'automate ci-dessus
4. Proposer un programme simulant l'automate ci-dessus

On considère l'alphabet = $\{0, 1, 2, \dots, 9\}$

Expression régulière = $(1|2|\dots|9)(0|1|2|\dots|9)^*|0$



E_0 = état initial E_1 et E_2 sont des états terminaux

Etat	0	1	2	3	4	5	6	7	8	9
E_0	E_1	E_2	E_2	E_2	E_2	E_2	E_2	E_2	E_2	E_2
E_1	-	-	-	-	-	-	-	-	-	-
E_2	E_2	E_2	E_2	E_2	E_2	E_2	E_2	E_2	E_2	E_2

Le programme ci-dessous représente l'automate ci-dessus.

```

def Etat0(chaine):
    if len(chaine)==0:
        return False
    elif chaine[0]=='0':
        return Etat1(chaine[1:])
    elif chaine[0]>='1' and chaine[0]<='9':
        return Etat2(chaine[1:])
    else:
        return False
# _____ #

def Etat1(chaine):
    if len(chaine)==0:
        return True
    else:
        return False
# _____ #

def Etat2(chaine):
    if len(chaine)==0:
        return True
    elif chaine[0]>='0' and chaine[0]<='9':
        return Etat2(chaine[1:])
    else:
        return False
# _____ #
  
```

Etude de cas 2 :

Nous allons modéliser une expression régulière qui dénote tous les nombres possédant une virgule

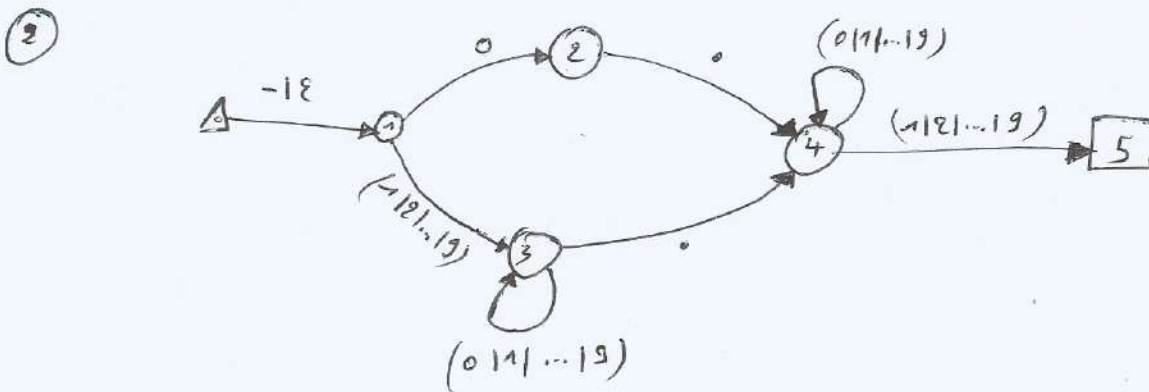
Exemples :

- 1,8
- 0,89
- -12,78

Pour cela il va falloir

1. Proposer une Expression régulière
2. Traduire l'ER en un automate
3. Déterminer l'automate ci-dessus
4. Proposer un programme simulant l'automate ci-dessus

① $(-|\epsilon) \left(((1|2|\dots|9)(0|1|\dots|9)^* | 0) \cdot (0|1|\dots|9)^* (1|2|\dots|9) \right)$



③