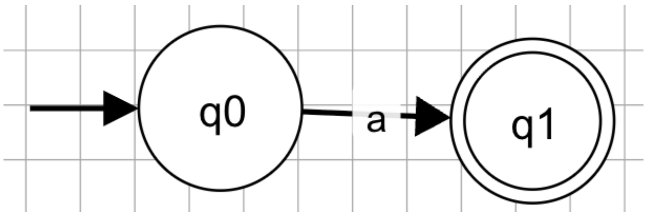
Test fonction union:

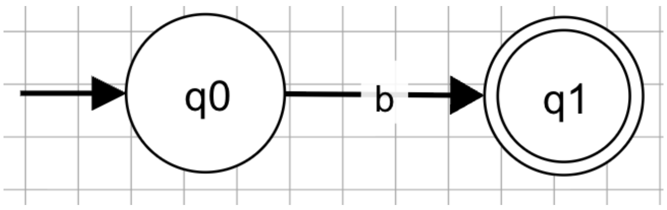
Test 1 :

Entrée :

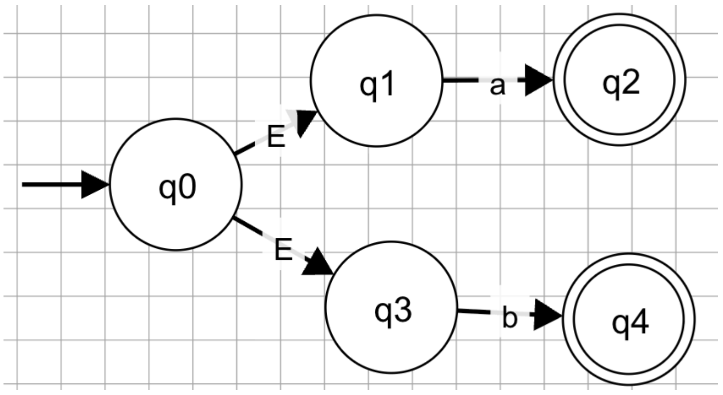
a :



b :



Sortie :



Code :

a = automate("a")

b = automate("b")

u = union(a, b)

print(u)

Sortie :

Automate ((a)+(b))

Nombre d'états 5

Etats finals [2, 4]

Transitions:

(1, 'a'): [2]

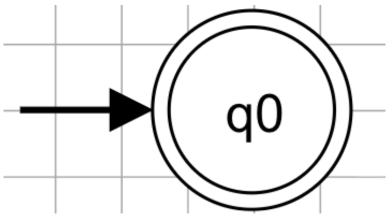
(3, 'b'): [4]

(0, 'E'): [1, 3]

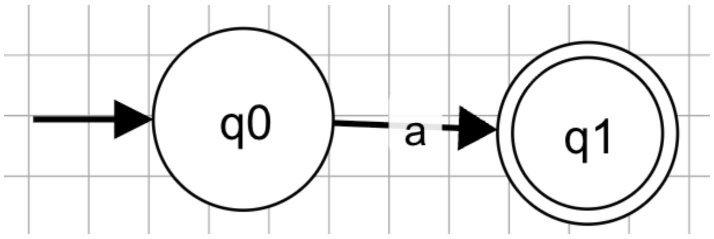
Test 2 :

Entrée :

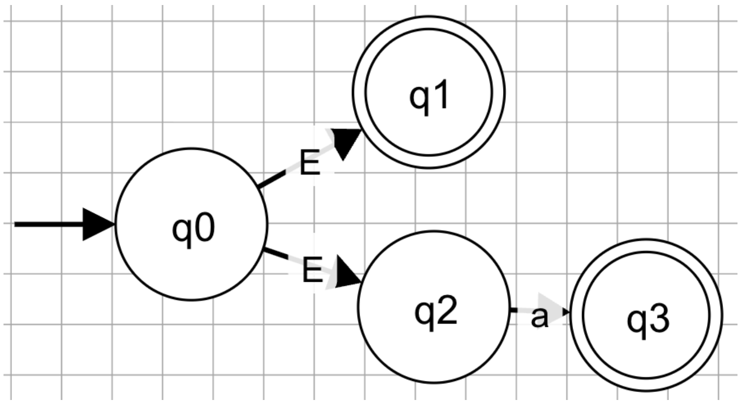
a :



b :



Sortie :



code :

a = automate("E")

b = automate("a")

u = union(a, b)

print(u)

Sortie :

Nombre d'états 4

Etats finals [1, 3]

Transitions:

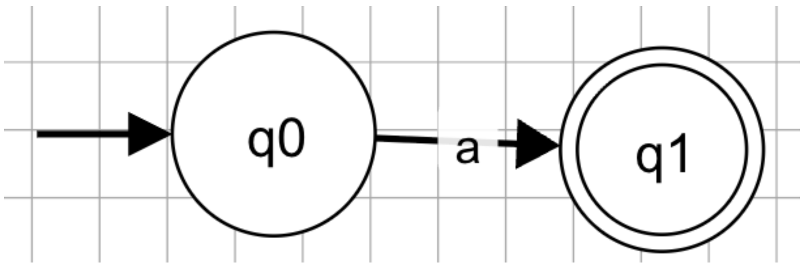
(2, 'a'): [3]

(0, 'E'): [1, 2]

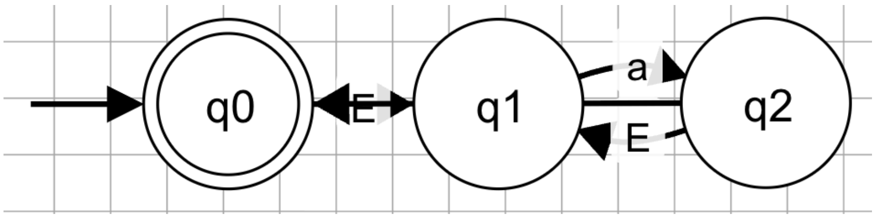
Test etoile

Test 1 :

a :



Sortie :



Code :

a = automate("a")

e = etoile(a)

```
print(e)
```

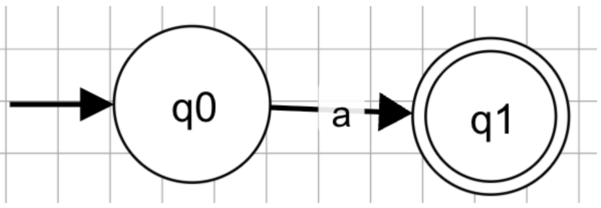
Sortie :

```
Automate ((a))*

Nombre d'états 3

Etats finals [0]

Transitions:

(1, 'a'): [2]

(0, 'E'): [1]

(2, 'E'): [1, 0]
```

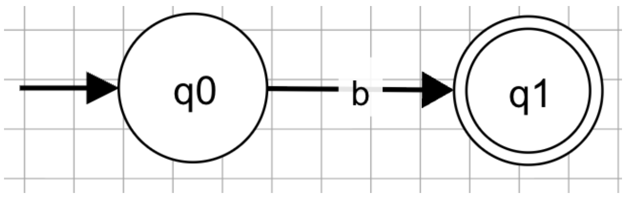
Test concatenation :

Test 1 :

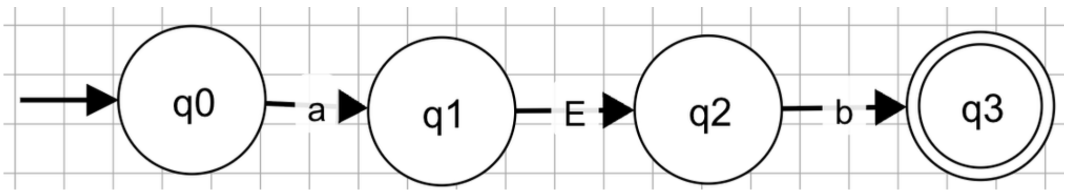
a :



b :



Sortie :



Code :

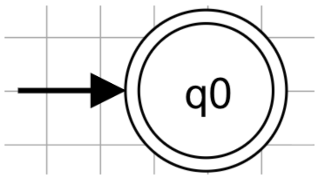
```
a = automate("a")

b = automate("b")

c = concatenation(a, b)

print(c)
```

Sortie :

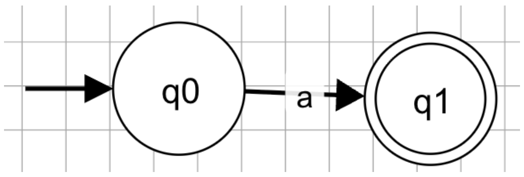
```
utomate ((a).(b))

Nombre d'états 4

Etats finals [3]

Transitions:

(0, 'a'): [1]

(2, 'b'): [3]

(1, 'E'): [2]
```

Test 2 :

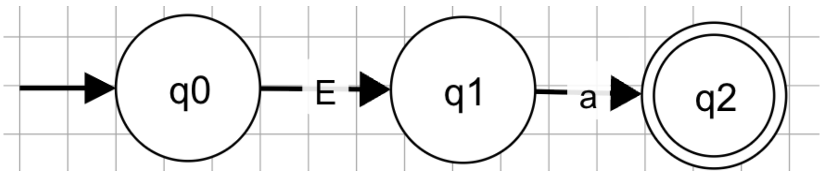
a :



b :



Sortie :



Code :

```
a = automate("E")

b = automate("a")

c = concatenation(a, b)

print(c)
```

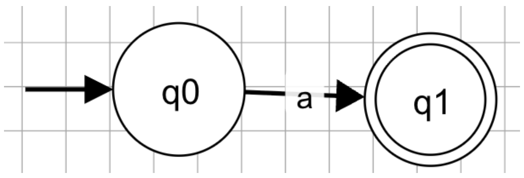
Sortie :

```
Automate ((E).(a))

Nombre d'états 3

Etats finals [2]

Transitions:

(1, 'a'): [2]

(0, 'E'): [1]
```

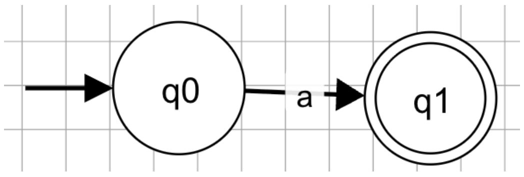
Test determinisation

Test 1 : automate deja deterministe à 2 états

a :



Sortie :



Code :

```
a = automate("a")

print("AVANT determinisation")

print(a)
```

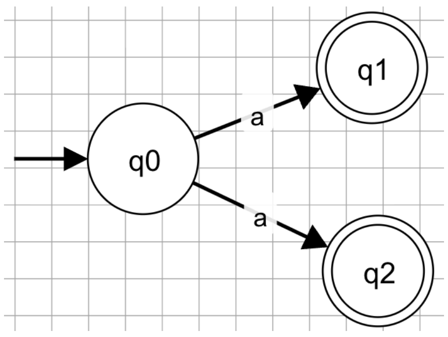
```
a_det = determinisation(a)

print("APRES determinisation")

print(a_det)
```

Sortie :

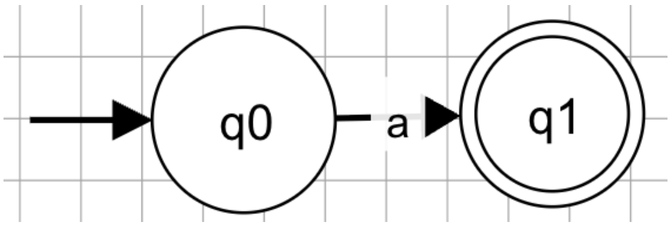
```
AVANT determinisation

Automate (a)

Nombre d'états 2

Etats finals [1]

Transitions:

(0, 'a'): [1]

********************************

APRES determinisation

Automate (a)

Nombre d'états 2

Etats finals [1]

Transitions:

(0, 'a'): [1]
```

Test 2. : automate non déterministe

a :



Sortie :



Code :

```
a = automate("O")
a.name = "(a_nd)"
a.n = 3
a.final = [1, 2]
a.transition = {
   (0, 'a'): [1, 2]}
print("AVANT determinisation")
print(a)
a_det = determinisation(a)
print("APRES determinisation")
print(a_det)
```

Sortie :

```
AVANT determinisation
Automate (a_nd)
Nombre d'états 3
Etats finals [1, 2]
Transitions:
(0, 'a'): [1, 2]
********************************
APRES determinisation
```

Automate (a_nd)

Nombre d'états 2

Etats finals [1]

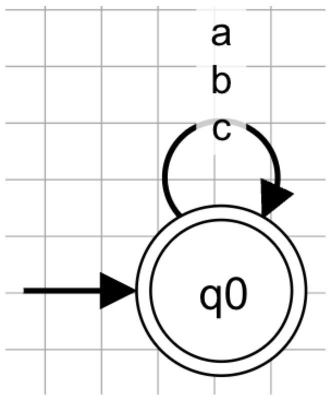
Transitions:

(0, 'a'): [1]

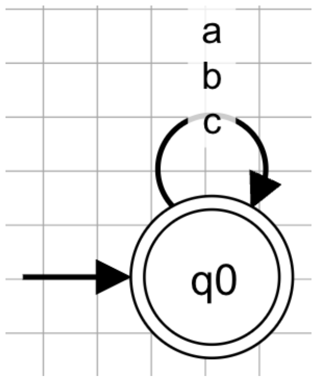
Test completion

Test 1 : Automate deja complet (a + b + c)*

a :



Sortie :



Code :

a = automate("0")

a.name = "(a+b+c)*"

a.alphabet = ["a", "b", "c"]

```
a.n = 1

a.final = [0]

a.transition[(0, "a")] = [0]

a.transition[(0, "b")] = [0]

a.transition[(0, "c")] = [0]

print("AVANT completion")

print(a)

a_comp = completion(a)

print("APRES completion")

print(a_comp)
```

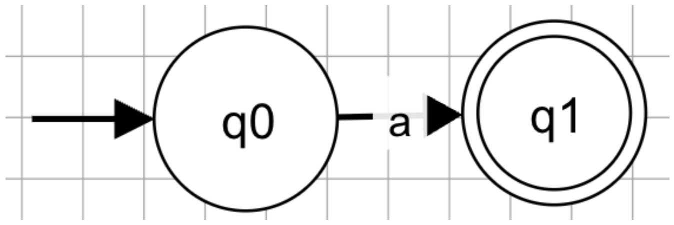
Sortie :

```
AVANT completion

Automate (a+b+c)*

Nombre d'états 1

Etats finals [0]

Transitions:

(0, 'a'): [0]

(0, 'b'): [0]

(0, 'c'): [0]

*******************************

APRES completion

Automate (a+b+c)*

Nombre d'états 1

Etats finals [0]

Transitions:
```

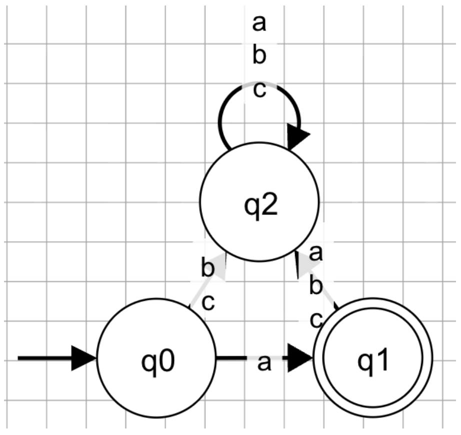
```
(0, 'a'): [0]

(0, 'b'): [0]

(0, 'c'): [0]
```

Test 2 : automate non complet, alphabet = {a, b, c}

a :



Sortie : q2 sert de poubelle



Code :

```
a = automate("a")

print("AVANT completion")

print(a)

a = completion(a)

print("APRES completion")

print(a)
```

Sortie :

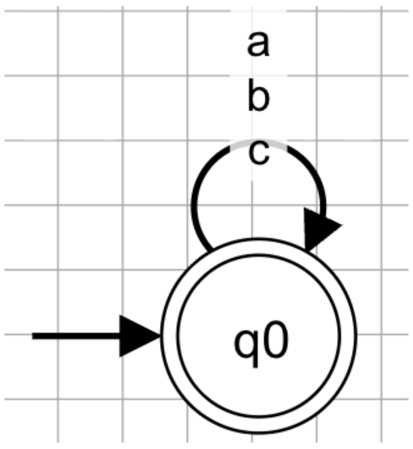
```
AVANT completion
```

```
Automate (a)

Nombre d'états 2

Etats finals [1]

Transitions:

(0, 'a'): [1]

********************************

APRES completion

Automate (a)

Nombre d'états 3

Etats finals [1]

Transitions:

(0, 'a'): [1]

(0, 'b'): [2]

(0, 'c'): [2]

(1, 'a'): [2]

(1, 'b'): [2]

(1, 'c'): [2]

(2, 'a'): [2]

(2, 'b'): [2]

(2, 'c'): [2]
```

Test egal

Test 1 : 2 automates égaux, (a+b+c)*

```
a1 = a2 :
```



Code :

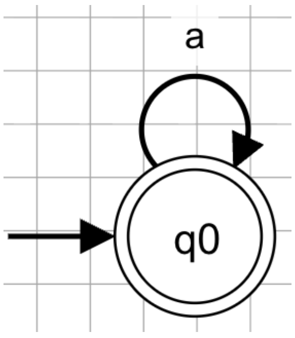
```
a1 = automate("O")

a1.name = "(a+b+c)*"

a1.alphabet = ["a", "b", "c"]

a1.n = 1

a1.final = [0]

a1.transition[(0,"a")] = [0]

a1.transition[(0,"b")] = [0]

a1.transition[(0,"c")] = [0]

a2 = automate("O")

a2.name = "(a+b+c)*"

a2.alphabet = ["a", "b", "c"]

a2.n = 1

a2.final = [0]

a2.transition[(0,"a")] = [0]

a2.transition[(0,"b")] = [0]

a2.transition[(0,"c")] = [0]

if egal(b1, b2) == True :

    print(“Egaux”)

Else :
    print(“Non egaux”)
```

Sortie :

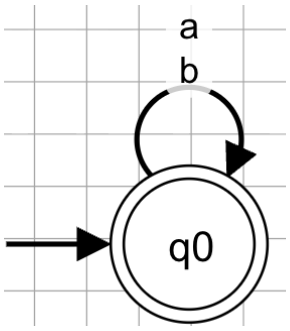
```
Egaux
```

Test 2 : 2 automates diffèrents,

b1 :



b2 :



Code :

```
b1 = automate("O")
b1.name = "a*"
b1.alphabet = ["a", "b"]
b1.n = 1
b1.final = [0]
b1.transition[(0,"a")] = [0]
b1.transition[(0,"b")] = [0]   # accepte b → langage plus grand
b2 = automate("O")
```

```
b2.name = "a*"

b2.alphabet = ["a", "b"]

b2.n = 1

b2.final = [0]

b2.transition[(0,"a")] = [0]

b2 = completion(b2)  # pour être comparable

if egal(b1, b2) == True :

    print("Egaux")

Else :

    print("Non egaux")
```

Sortie :

```
Non egaux
```