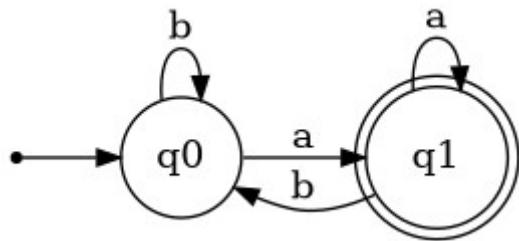


Traçage automate avec python3 :

Avec bibliothèque « automata »



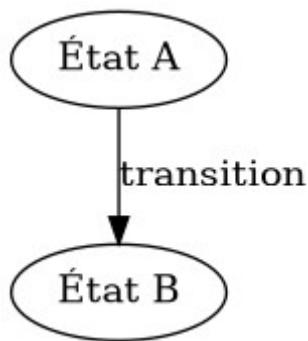
Code :

```
from automata.fa.dfa import DFA

# Définition de l'automate
dfa = DFA(
    states={'q0', 'q1'},
    input_symbols={'a', 'b'},
    transitions={
        'q0': {'a': 'q1', 'b': 'q0'},
        'q1': {'a': 'q1', 'b': 'q0'}
    },
    initial_state='q0',
    final_states={'q1'}
)

# Tester quelques chaînes
print(dfa.accepts_input('a')) # True
print(dfa.accepts_input('b')) # False
print(dfa.accepts_input('aa')) # True
print(dfa.accepts_input('ab')) # True
```

Avec bibliothèque « graphviz »



Code :

```
from graphviz import Digraph

# Création du graphe orienté
dot = Digraph(comment='Exemple avec 5 états')

# Ajout des états
dot.node('A', 'État A')
dot.node('B', 'État B')
dot.node('C', 'État C')
dot.node('D', 'État D')
dot.node('E', 'État E')

# Ajout des transitions
dot.edge('A', 'B', label='A → B')
dot.edge('A', 'C', label='A → C')
dot.edge('B', 'D', label='B → D')
dot.edge('C', 'D', label='C → D')
dot.edge('D', 'E', label='D → E')

# Génération et affichage du graphe
dot.render('graphe_5_etats', format='png', view=True)
```