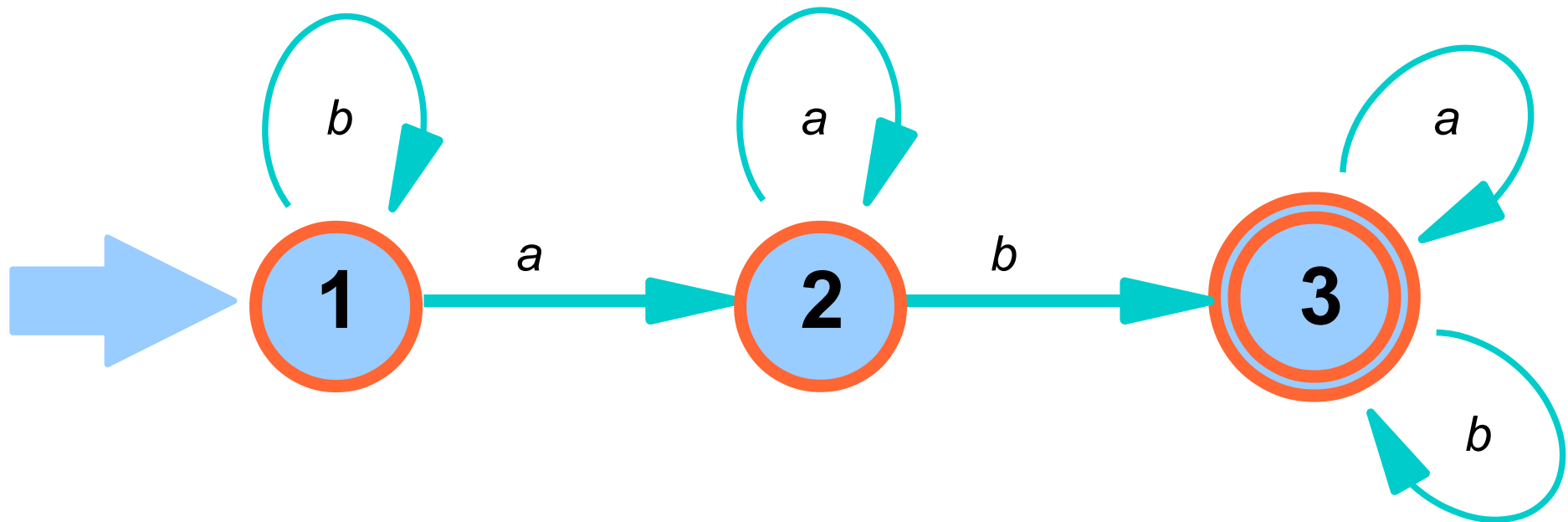


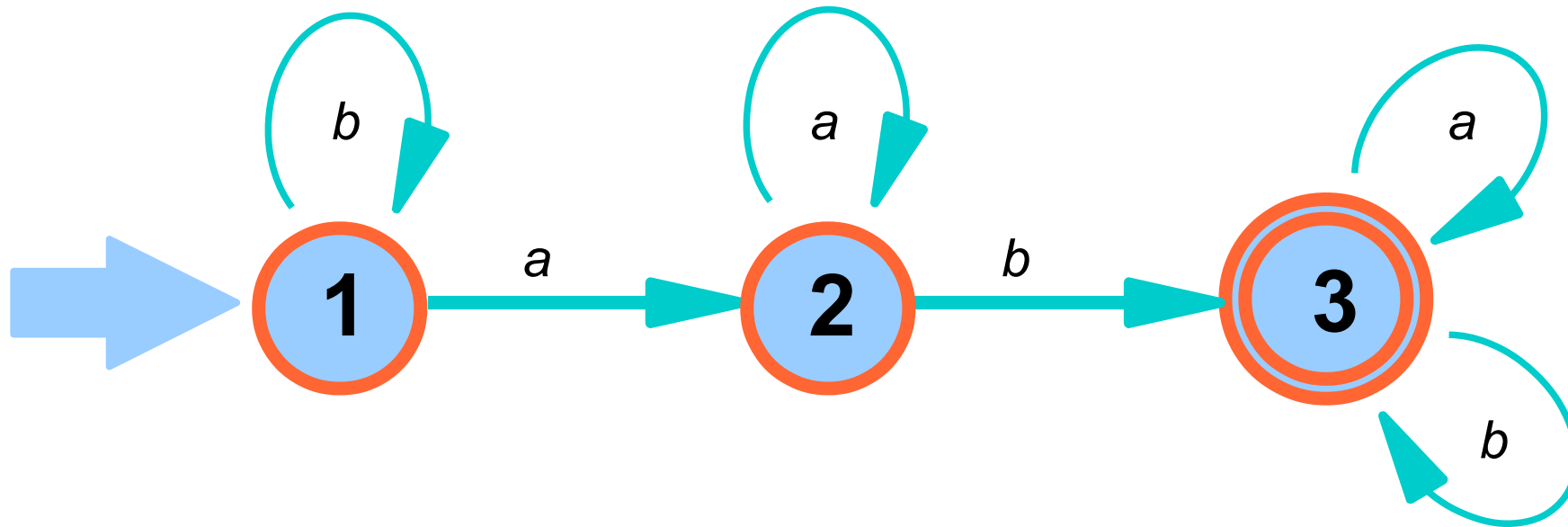
Représentations d'un automate :

Par un graphe valué :



Les sommets sont les états, les transitions sont des arcs valués par la lettre lue. On précise état initial et états finaux.

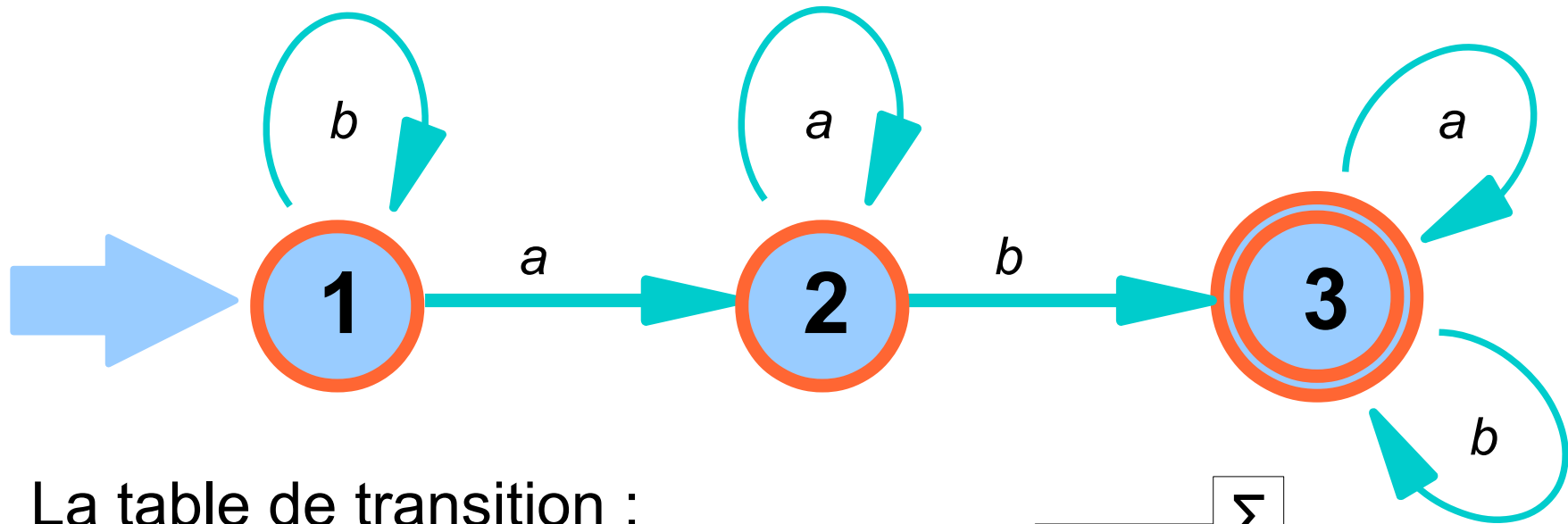
Représentations d'un automate :



Mathématiquement :

- $\Sigma = \{a,b\}$ Un ensemble
- $Q = \{1,2,3\}$ Un ensemble ou seulement l'entier $n=3$
- $q_0 = 1$ Un entier
- $A = \{3\}$ Une partie de l'ensemble Q
- T Une fonction de $Q \times \Sigma \rightarrow Q$

Représentations d'un automate :



La table de transition :

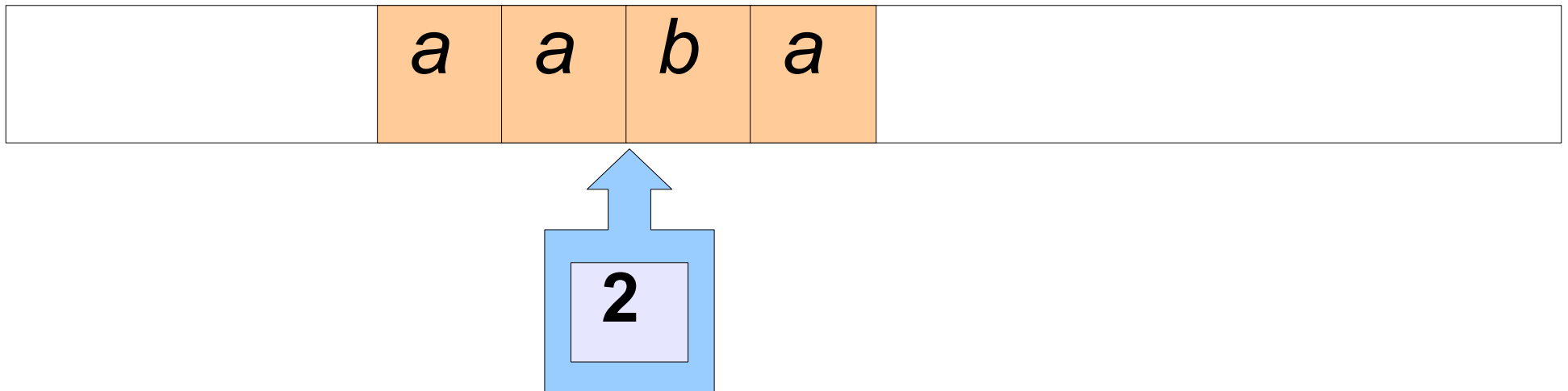
| Q \ Σ | Σ | |
|--------------|----------|---|
| | a | b |
| 1 | 2 | 1 |
| 2 | 3 | 3 |
| 3 | 3 | 1 |

Annotations:

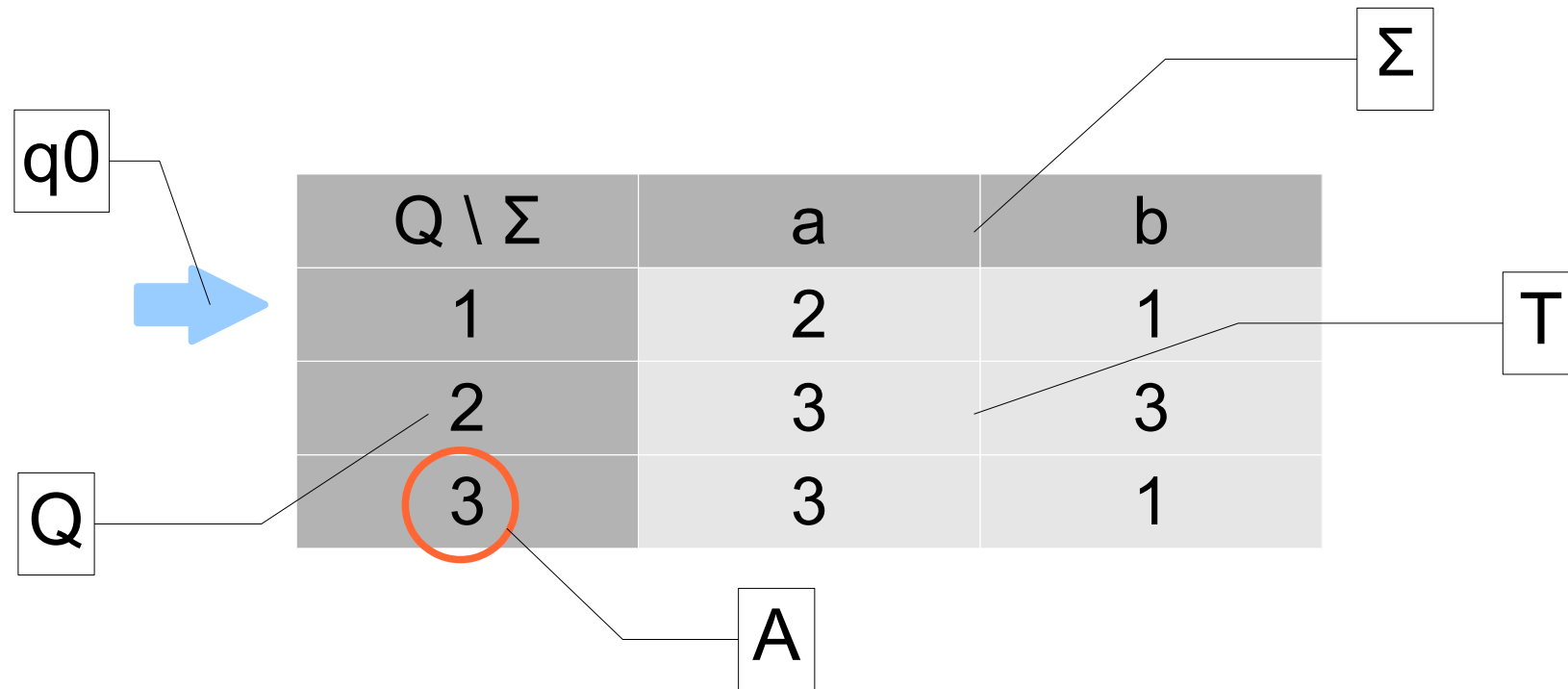
- q_0 points to state 1.
- Q points to the set of states {1, 2, 3}.
- A points to state 3.
- T points to the transition from state 2 on input 'b' to state 3.

Représentations d'un automate :

Une machine à ruban

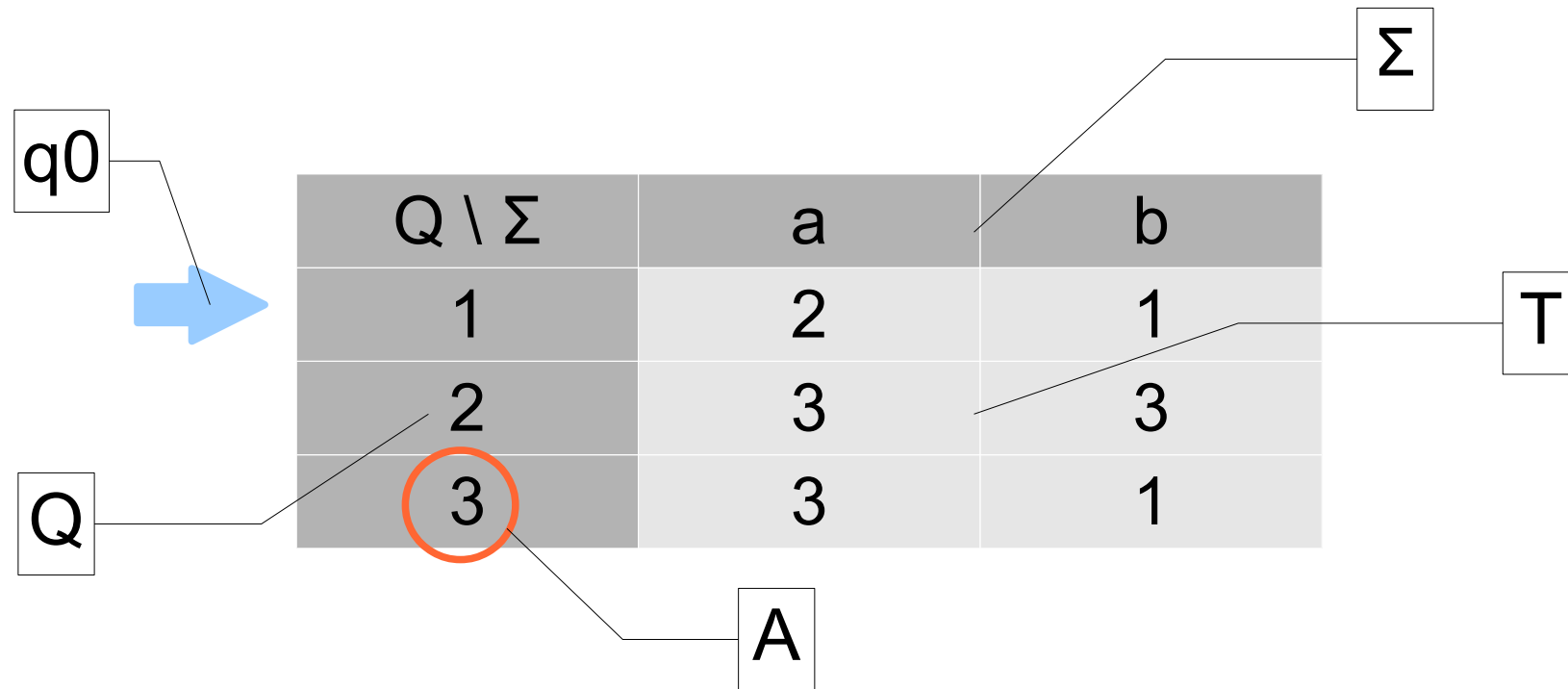


Ne permet que l'illustration de la lecture d'un mot.



En caml (par exemple) par un type produit :

```
type AFD = {
  Sigma : char list; (* l'alphabet *)
  N : int; (* Q est l'ensemble {1..N} *)
  Init : int; (* l'état initial *)
  A : int list; (* les états acceptants *)
  T : (int*char) list vect (* le graphe de transition
    représenté par ses listes d'adjacences valuées *)};;
```



```
let A= {Sigma=[`a`;`b`]; N=3; Init =1; A =[3];
      T = [[(`a`,2);(`b`,1)] ; [(`a`,3);(`b`,3)] ; [(`a`,3);(`b`,1)]  ]];;
```

```
A : AFD = {Sigma = [`a`; `b`]; N = 3; Init = 1; A = [3];
          T = [[[`a`, 2; `b`, 1]; [`a`, 3; `b`, 3]; [`a`, 3; `b`, 1]]]}
```