

ACTIVITÉ N° 2 - LECTURE D'UN MOT PAR AUTOMATE FINI NON DÉTERMINISTE

Piste Verte

L'objectif de ce TP est d'implémenter la lecture d'un mot par un Automate Fini Non Déterministe (AFN dans toute la suite), sans ϵ -transition.

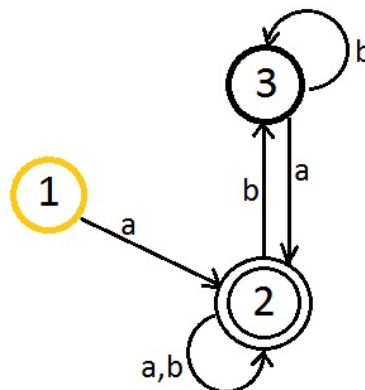
1. Introduction - Un type pour les AFN

Dans un premier temps, nous supposons qu'il n'y a pas d' ϵ -transitions. Pour représenter ces automates, nous introduisons les types suivants, semblables à ceux des automates finis déterministes du TP précédent.

```
type etatN = {acceptN : bool ; tN : char -> int list};;
```

```
type afn = {sigmaN : char list ; (* l'alphabet *)
  nN : int ; (* Q est l'ensemble {1..N} *)
  initN : int list ; (* les états initiaux *)
  eN : int -> etatN};;
```

L'automate qui nous servira d'exemple est le suivant :



2. Lecture d'un mot par un AFN

1. Écrire une fonction `transitN` : `afn * int * char -> int list` qui prend en entrée un automate, un état q et un caractère a . La fonction renvoie la liste des états q' tels que (q, a, q') est une transition de l'automate. La fonction lève une exception s'il n'y a pas de transition.

Exemple :

```
transitN (an1, 1, 'b') ; ;
Exception PasTransition
```

```
transitN (an1, 2, 'b') ; ;
- : int list = [2 ; 3]
```

2. (a) Écrire une fonction `lire_auxN` qui prend en entrée un automate, une liste d'états ℓ et un mot. La fonction renvoie la liste des états atteints à l'issue de toutes les lectures possibles du mot w par l'automate à partir des états de la liste ℓ . Comme dans le TP précédent, la fonction renvoie un 0 s'il n'y a pas de transitions.
- (b) En déduire une fonction `lireN` : `afn -> string -> int list` qui effectue toutes les lectures d'un mot w dans un automate à partir des états initiaux.

Exemple :

```
let l2N = lireN an1 ; ;
(*l2N : string -> int list = <fun>*)

List.map l2N ["a" ; "ab" ; "abb" ; "abba" ; "b"] ; ;
(*- : int list list = [[2] ; [2 ; 3] ; [2 ; 3 ; 3] ; [2 ; 2 ; 2] ; [0]]*)
```

3. (a) Écrire une fonction `accept_auxN` qui prend en entrée un automate et une liste d'états, et renvoie `true` si la liste contient un état acceptant.
- (b) En déduire une fonction `acceptN` qui teste la reconnaissance d'un mot par un automate.

```
accept an1 "abba" ; ;
(*- : bool = true*)
```