Institut National Universitaire Champollion

TP 2 - Lecture d'un mot par un automate

Théorie des langages
L3 INFO - Semestre 6

Activité n° 2 - Lecture d'un mot par Automate Fini Non Déterministe

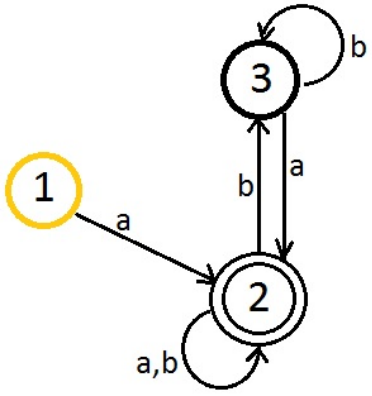
Piste Bleue

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.

1. Comment peut-on adapter les types `etat` et `afd` introduits pour les AFD aux AFN ?
   Définissez un type `etatN` et un type `afn`. Faites les valider avant de passer à la suite !
2. Définissez l'automate $an1$ ci-contre, qui servira à tester vos fonctions.



2. Lecture d'un mot par un AFN

Partie 1. Une fonction de transition Écrire une fonction `transitN : afn * int * char -> int list` qui a le comportement suivant :

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

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

Partie 2. Lecture d'un mot par un automate Écrire une fonction `lireN : afn -> string -> int list` qui effectue toutes les lectures d'un mot w dans un automate à partir des états initiaux.

```
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]]*)
```

Partie 3. Bilan En déduire une fonction `acceptN` qui teste la reconnaissance d'un mot par un automate.

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