Institut National Universitaire Champollion

TP 2 - Lecture d'un mot par un automate

Théorie des langages
L3 INFO - Semestre 6

Activité n° 1 : Lecture d'un mot par un Automate Fini Déterministe

Piste Rouge

1. Rappel : Le type afd

Le type etat Pour définir les automates finis déterministes, nous introduisons tout d'abord le type produit suivant :

```
type etat = {accept : bool ; t : char -> int} ;;
```

Ce type permet de décrire les états de l'automate. Le booléen `accept` est égal à `true` si l'état est acceptant, `false` sinon. La fonction `t` décrit les transitions à partir de cet état.

Le type afd Nous représenterons les automates finis déterministes par le type suivant :

```
type afd = {sigma : char list ; nQ : int ; init : int ; e : int -> etat} ;;
```

Ici, `sigma` désigne l'alphabet, `nQ` est le nombre d'états numérotés de 1 à nQ, `init` est le numéro de l'état initial et enfin, e est une fonction qui à chaque entier désignant un état associe sa description de type `etat`.

2. Lecture d'un mot par un automate

Écrire une fonction `accepte : afd -> string -> bool` qui teste si un mot est reconnu ou non par un automate.

```
let ac1 = accepte a1 ;;
val ac1 : string -> bool = <fun>

List.map ac1 ["abba";"bbaaa";"bbaaba";"ba";"ab";""] ;;
- : bool list = [true; false; true; false; true; false]

accepte a2 "babb" ;;
- : bool = false
```

Définir un automate qui reconnaît les mots de la forme $a^n b$. Tester vos fonctions.