

TP N° 3 - SUPPRIMER LES ε -TRANSITIONS D'UN AFN Piste Noire

Cette fois, nous considérons des automates finis non déterministes pourvus d' ε -transitions. Nous emploierons le caractère 'é' pour désigner ces transitions. Le fichier Caml vous propose deux exemples d'automates. L'automate *an2* est le même automate que celui de l'animation "Déterminer un automate" disponible sur Moodle.

L'objectif de cette partie est de supprimer les ε -transitions d'un AFN_{ε} .

Calcul des clôtures

Écrire une fonction `cloture` : `afn -> int -> int list` qui détermine la clôture de l'état *i*.

Les tests

```
let clotureDansAN1 = cloture an1 ;; (* clotureDansAN1 : int -> int list *)
let clotureDansAN2 = cloture an2 ;; (* clotureDansAN2 : int -> int list *)

List.map clotureDansAN1 [1;2;3;4] ;;
(*int list list = [[2; 3; 1; 4]; [4; 2; 1; 3]; [2; 3; 1; 4]; [4; 2; 1; 3]] *)

List.map clotureDansAN2 [1;2;3;4] ;;
(*int list list = [[2; 3; 1; 4]; [4; 2; 1; 3]; [2; 3; 1; 4]; [4; 2; 1; 3]] *)
```

États acceptants

Écrire une fonction `nouvelEtatAcceptant` : `afn -> int -> bool` qui prend un automate *an* et un état *q* et renvoie vrai si *q* sera un état acceptant dans l'automate *an* privé de ses ε -transitions.

Les tests

```
let nouveauxAcceptantsAN2 = nouvelEtatAcceptant an2 ;;
(*nouveauxAcceptantsAN2 : int -> bool*)

List.map nouveauxAcceptantsAN2 [1;2;3;4] ;;
(*bool list = [false; true; true; false]*)
```

Transitions étendues

Écrire une fonction `transitionsEtendues` : `afn -> char -> int -> int list` qui prend en paramètres un automate, un caractère c et un état q et détermine les transitions étendues, c'est à dire la liste des états q' tels que (q, c, q') est une transition dans le nouvel automate sans ε -transition.

Les tests

```
let tN_AN2_a = transitionsEtendues an2 'a' ;; (*tN_AN2_a : int -> int list*)
List.map tN_AN2_a [1;2;3;4] ;;
(* int list list = [[2]; [2]; [2]; [2]]*)
```

```
let tN_AN2_b = transitionsEtendues an2 'b' ;; (*tN_AN2_b : int -> int list*)
List.map tN_AN2_b [1;2;3;4] ;;
(* : int list list = [[]; [2; 4]; [4; 2]; [4]]*)
```

Construction de l' AFN équivalent

Écrire une fonction `nouvelAuto` : `afn -> afn` qui prend un AFN_ε et renvoie l' AFN équivalent.

À vous de tester !