TP 8— Automates et Eleusis

http://www.liafa.jussieu.fr/~nath/tpcaml/tp8/tp8.pdf

Les questions sont les bienvenues et peuvent être envoyées à nathanael.fijalkow@gmail.com.

1 Le plus petit mot

Nous nous intéressons au problème suivant: étant donné un langage rationnel L, déterminer la longueur minimale d'un mot de L. Nous envisagerons deux situations: lorsque L est donné par un automate déterministe, et lorsque L est décrit par une expression rationnelle. L'algorithme devra détecter le cas où L est vide.

1.1 Expressions rationnelles

▷ **Question 1.** Définissez par induction structurelle une fonction des expressions rationnelles qui détermine si un langage est vide. ◁

On définit le type des expressions rationnelles :

```
type expr = Vide
 | Epsilon
 | Lettre of char
 | Union of expr * expr
 | Concat of expr * expr
 | Etoile of expr ;;
```

▷ **Question 2.** Écrire la fonction `est_vide` qui calcule la fonction précédente. ◁

▷ **Question 3.** Définissez par induction structurelle une fonction des expressions rationnelles vers les entiers qui calcule le longueur du plus court mot de L si le langage n'est pas vide, et ∞ sinon. ◁

▷ **Question 4.** Écrire la fonction `longueur_mot_min` qui calcule la fonction précédente. Elle retournera un objet de type `int option`. ◁

1.2 Automates déterministes

```
type automate =
 { taille :  int ;
   initial :  int ;
   transitions :  (char * int) list vect ;
   final :  bool vect } ;;
```

▷ **Question 5.** Écrire une fonction `est_vide_auto` qui détermine si le langage de l'automate est vide. Écrire une fonction `longueur_mot_min_auto` qui calcule la longueur du mot minimal accepté par l'automate. Elle retournera un objet de type `int option`. ◁

▷ **Question 6.** Écrire une fonction `langage_auto` qui retourne la liste de tous les mots de taille minimale reconnu par l'automate. Quelle est sa complexité? ◁

1.3 Les automates pour ne pas réfléchir

Un renard se cache dans une des cinq tanières, numérotées de 1 à 5. Chaque nuit, il doit changer de tanière, mais n'a le temps d'atteindre que l'une des deux tanières les plus proches (par exemple, depuis la tanière 2, le renard doit aller soit en 1, soit en 3). Depuis 1, il doit aller en 2, et depuis 5, il doit aller en 4. Chaque jour, le chasseur choisit une tanière et tue le renard s'il s'y trouve. Le chasseur a t-il une stratégie lui permettant de tuer le renard ? Combien de jours faut-il pour cela ?

2 Eleusis

La page d'homonymie de Wikipédia vous apprendra les diverses origines et utilisations du mot Eleusis. On s'intéresse au jeu de cartes (par ailleurs édité pour le grand public). Le principe est le suivant : un joueur, le Maître, choisit une règle (ici, un automate déterministe). Les autres joueurs doivent deviner cette règle. Pour cela, chacun à leur tour, ils vont proposer une carte, 0 ou 1. Si l'automate possède une transition depuis l'état

courant pour la lettre proposée, il répond "Accepté" et le nouvel état courant est mis à jour. Sinon, il répond "Rejeté" et conserve l'état courant. Évidemment, l'état courant est conservé secret à tout moment par le Maître.

Un joueur gagne si à partir d'un moment, toutes ses cartes sont acceptées (ce qui est sensiblement différent d'avoir "deviné" la règle).

Vous allez jouer à Eleusis, seul (le Maître contre vous). Avantage non négligeable : vous connaissez la taille de l'automate, qui est au plus n. Il s'agit de trouver une stratégie, simple et rapide, pour que toutes les cartes que vous proposez soient acceptées.

▷ **Question 7.** Jouer (recopier le code ci-contre, programmer un automate et jouer avec l'automate du voisin) ! ◁

```
let jouer auto =
 let rec boucle etat_courant =
 print_string "Proposer une carte (0 ou 1) " ;
  try
   let carte = (read_line ()).[0] in
   let nv_etat = assoc carte auto.transitions.(etat_courant) in
   print_string "Accepté \n" ; print_newline () ; boucle nv_etat
  with
   | Not_found -> print_string "Rejeté \n" ;
   print_newline() ;
   boucle etat_courant
   | End_of_file -> print_string "Fini \n" ;
  in boucle auto.initial ;;
```

▷ **Question 8.** Existe t-il une stratégie gagnante ? ◁

▷ **Question 9.** Écrire une fonction `strategie`. Elle prend en entrée une liste de couples `char * bool`, le premier élément étant la carte proposée, et le second la réponse du Maître, et retourne une nouvelle carte à jouer. ◁

▷ **Question 10.** Évaluer la complexité de votre stratégie : si on suppose que l'automate est de taille n, combien de cartes au plus sont refusées? quelle est la complexité de calcul du prochain coup? ◁

Merci à Axel Haddad et à Denis Kuperberg pour leurs contributions à l'étude d'Eleusis.