Caml. Fractales et Automates

Dominique Michelucci, École des Mines de Saint-Étienne

Un automate déterministe est défini par son état (de type int pour simplifier) initial, une fonction de transition et une valuation. La fonction de transition donne, pour l'état courant de l'automate et une transition possible (un caractère par exemple), le nouvel état de l'automate. La fonction de valuation donne, pour un état de l'automate, une valeur (de type int pour simplifier). Cette valeur est celle calculée par l'automate pour une entrée (chaine de caractères) donnée.

```
type Etat==int;;
type AD= { initial: Etat;
           arcs : Etat-> char->Etat;
           vals : Etat->int; (* numero etat -> valeur *)
         };;
```

Définisser la fonction: executer, qui simule le comportement de l'automate passé en argument. La valeur rendue est la valuation de l'état final dans lequel se trouve l'automate, après qu'il ait parcouru son entrée, une chaine de caractère.

1 La suite de Thue Morse $T(n)$

La suite de Thue Morse est une suite de 0 et de 1, qui a la propriété de ne jamais contenir trois fois de suite la même séquence. Il s'avère qu'elle est automatique, en ce sens qu'il existe un automate qui, après lecture de l'expansion en binaire de n, rend le n ième terme de la suite. Cet automate a deux états : l'état initial i et l'état a. S'il est dans l'état i, alors un 0 le fait passer en i et un 1 en a; s'il est dans l'état a, alors un 0 le fait passer en a et un 0 en i. Les valuations de a et i sont respectivement 1 et 0.

Définissez cet automate, une fonction convertissant un entier positif en son expansion binaire inversée (par exemple : $4 \to 001$, $10 \to 0101$), une fonction mot: `AD -> int -> int -> string` telle que :

```
mot auto_thuemorse 0 31;;
- : string = "01101001100101101001011001101001"
```

et calculez les premiers termes de la suite. Quelles remarques faites-vous sur cette suite ?

2 Pliage de papier $P(n)$

Pliez une feuille de papier en 2, en 4, etc, toujours dans le même sens; dépliez; les plis sont en creux (un 0) ou en bosse (un 1); la suite obtenue est elle aussi automatique. Si vous dépliez de telle façon que les plis forment des angles droits, vous obtenez la courbe dite du pliage, ou du serpent (ou peut-être du dragon: je confonds...).

```
mot auto_pliage 0 16;;
- : string = "1101100111001001 1"
```

On admettra que $P(n)$ est 1 moins le chiffre précédant le 0 le plus à droite (de poids le plus faible) dans l'écriture en base 2 de n (l'écriture se fait ici dans le sens habituel). Par exemple : $13 = (1\dot{1}01) \to 0$, $9 = (1\hat{0}01)_2 \to 1$. Quand la chaine ne contient pas de zéro ? On peut toujours ajouter deux 0 devant un entier, sans modifier sa valeur...

Un automate solution a 4 états: i, a, b, c. Les transitions sont : $i.0 = a$, $i.1 = i$, $a.0 = b$, $a.1 = c$, $b.0 = b.1 = b$, $c.0 = c.1 = c$, les valuations sont $i \to 1$, $a \to 1$, $b \to 1$ et $c \to 0$. Définissez cet automate, calculez les premiers termes de P.

Dessinez la courbe du pliage : se placer au centre de l'écran. Pour i de 0 à n, si $P(i) = 0$ alors avancer en tournant à droite de 90 degrés, sinon avancer en tournant à gauche de 90 degrés. Vous devriez obtenir une image similaire à la figure 1.

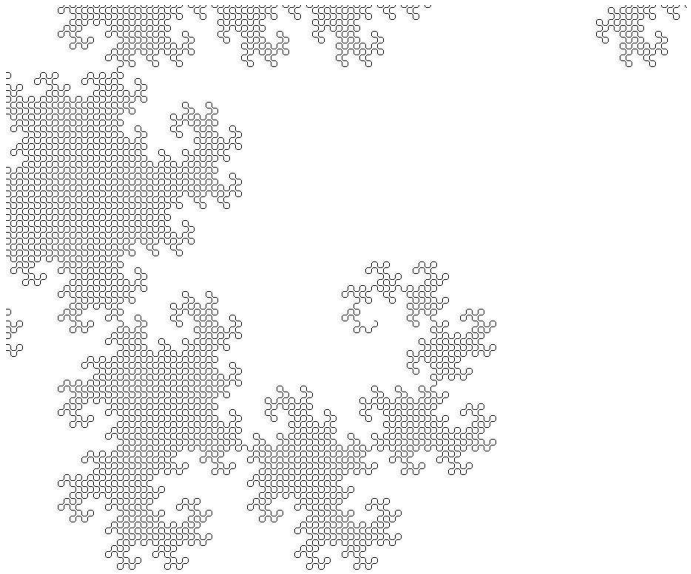

Figure 1: La courbe du pliage de papier

3 La suite de Von Koch $K(n)$

$K(n)$ est la parité du nombre de zéros consécutifs qui terminent lécriture en base 2 de n. Si n est impair, le nombre de 0 à la fin est nul, donc pair. D'où $P(2k+1) = 0$. A noter que le cas de $P(0)$ se discute, selon que l'on estime que 0 s'écrit 0 ou le mot vide : j'ai choisi la seconde possibilité.

Un automate solution a 4 états i, a, b, c, i est initial. Les transitions sont : $i.0 = a$, $i.1 = b$, $a.0 = i$, $a.1 = c$, $b.0 = b.1 = b$, $c.0 = c.1 = c$. Les valuations sont : $i \to 0$, $a \to 1$, $b \to 0$, $c \to 1$.

```
mot auto_koch 0 31;;
- : string = "00100010101000100010001010100010"
```

Définissez cet automate, calculer les premiers termes de la suite. La suite K permet de dessiner la suite de Von Koch : à l'étape n, avancer d'un pas, puis tourner de 60 ou 240 degrés selon que $K(n)$ vaut 0 ou 1. Dessinez la courbe de Von Koch (figure 2).

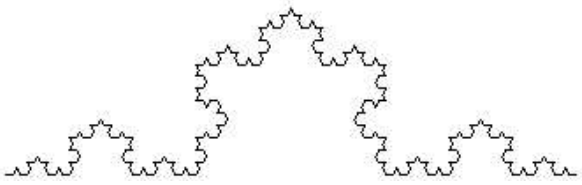

Figure 2: La courbe de Von Koch

```
#open "graphics";;
open_graph " 500x500";;

type Etat==int;;
type AD= { initial: Etat;
       arcs : Etat-> char->Etat;
       vals : Etat->int; };;

(* ecriture en base 2 , INVERSEE: 4 --> "001" *)
let rec enbase2 =function
      0 -> "" | 1 -> "1"
    | n -> (if (0=n mod 2) then "0" else "1") ^ (enbase2 (n/2));;

(* simule un automate; rend la valeur de l'etat final *)
let execute {initial=etat0; arcs=f; vals= v; }  mot=
    let n=string_length mot in
    let state= ref etat0 in
    for i=0 to n-1 do state := f !state mot.[i]
    done;
    v !state;;

(* convertit un entier (petit) en carateres *)
let int2char x= char_of_int (x+int_of_char `0`);;

(* convertit un automate en fonction: int->int *)
let fonction_automate automate i = execute automate (enbase2 i);;

(* rend les valeurs de l'automate pour les chaines expansions en base 2
   (inversee!) de i= i1...i2 *)
let mot automate i1 i2=
    let mot=create_string (i2-i1+1) in
    let fonc = fonction_automate automate in
    for i=i1 to i2 do mot.[i-i1] <- int2char (fonc i) done; mot;;

(* pliage(n) est 1 moins le chiffre precedant le 0 le plus a droite
   (de poids le plus faible) de l'ecriture usuelle en base 2 de n.
   Il y a tjrs un dernier 0, en
   supposant que les entiers sont precedes d'une infinite de 0  *)
exception ErreurPliage;;
let auto_pliage=
{   initial=0;
    arcs = ( fun
          0 `1` -> 0 | 0 `0` -> 1
        | 1 `0` -> 2 | 1 `1` -> 3
        | 2 _ -> 2 | 3 _ -> 3
        | _ _ -> raise ErreurPliage) ;
    vals = ( function
       0 -> 1 | 1 -> 1 | 2 -> 1 | 3 -> 0 | _-> raise ErreurPliage );
};;
let fpliage= fonction_automate auto_pliage;;
let mpliage= mot auto_pliage 0 128;;

let current_dx=ref 1;;
let current_dy=ref 0;;
let ll= ref 2;;

(* sens=1: on tourne a droite; sens=-1: on tourne a gauche *)
let tourner sens=
    let (x,y)=current_point() in
    let dx= !current_dx and dy= !current_dy in
    let dx2= sens*(-dy) and dy2=sens*dx in
    let xA=x+ !ll*dx and yA=y+ !ll*dy in
    let xB=xA+ !ll*(dx+dx2) and yB=yA+ !ll*(dy+dy2) in
    let xC=x+ !ll*2*(dx+dx2) and yC=y+ !ll*2*(dy+dy2) in
    begin lineto xA yA; lineto xB yB; lineto xC yC;
          current_dx := dx2; current_dy := dy2
    end ;;

(* on peut dessiner la courbe fractale du pliage de papier *)
let dessine_pliage n =
        clear_graph(); current_dx:=1; current_dy:=0;
```

```
        set_color black; moveto (size_x()/2)  (size_y()/2);
        for i=0 to n do (
            match (fpliage i) with
                  0 -> tourner 1
                | 1 -> tourner (-1)
                | _ -> failwith "argl" )
        done ;;

dessine_pliage 2048;;

(* Courbe de vonkoch *)
(* koch(n)= parite du nb de zeros consecutifs a la fin de l'ecriture
   en base 2 de n  (pour n > 0; pour n=0, ca se discute...) *)

exception ErreurKoch;;
let auto_koch=
{   initial=0;
    arcs = ( fun
          0 '1' -> 2 | 0 '0' -> 1
        | 1 '0' -> 0 | 1 '1' -> 3
        | 2 _ -> 2 | 3 _ -> 3 | _ _ -> raise ErreurKoch) ;
    vals = ( function x -> x mod 2 ); };;

let fkoch= fonction_automate auto_koch;;
let mkoch= mot auto_koch 0 128;;

(* Dessin de la courbe de von Koch
    l = ref vers la longueur dont on avance a chaque etape
    theta = angle en degres de la direction courante
        si koch(i) vaut 0, il est incremente de 60
        si koch(i) vaut 1, il est incremente de 240
    remarque: on ne se sert pas de koch(0)
*)
let pi= 4. *. (atan 1.);;
let deg2rad x= (float_of_int x) *. pi /. 180.;;
let l=ref 4.;;

let dessine_koch n=
    clear_graph();
    let theta=ref 0 in
    let x= ref 10.
    and y= ref  (0.3 *. (float_of_int (size_y())))  in
        moveto (int_of_float !x) (int_of_float !y);
    for i=1 to n  do
      x := !x +. !l *. (cos (deg2rad !theta));
      y := !y +. !l *. (sin (deg2rad !theta));
      lineto (int_of_float !x) (int_of_float !y);
      (* mise a jour de l'orientation theta : *)
      (match (fkoch i) with
          0 -> theta:= (!theta+60) mod 360
        | 1 -> theta:= (!theta+240)  mod 360
        | _ -> failwith "argl"
      )
     done ;;

dessine_koch 2048;;

(*  thuemorse( n) = parite du nombre de 1 dans
   l'ecriture en base 2 de n, pour n positif ou nul *)

exception ErreurThueMorse;;
let auto_thuemorse=
{   initial=0;
    arcs = ( fun
      0 '0' -> 0 | 0 '1' -> 1
    | 1 '1' -> 0 | 1 '0' -> 1
    | _ _ -> raise ErreurThueMorse) ;
    vals = ( function 0 -> 0 | 1 -> 1 | _-> raise ErreurThueMorse ); };;

let fthuemorse= fonction_automate auto_thuemorse;;
let mthuemorse= mot auto_thuemorse 0 128;;
```