

# Agrégation de Mathématiques option Informatique

## TP2

Loïc JEZEQUEL

Mardi 25 Octobre 2011

Ce TP a pour objectif de vous familiariser avec le module `Graphics` de Caml.

### Dessiner avec Caml

En utilisant le module `Graphics` de Caml il est possible de dessiner. On ouvre une fenêtre par la fonction suivante :

```
Graphics.open_graph : string -> unit
```

et on la ferme par la fonction suivante :

```
Graphics.close_graph : unit -> unit
```

Il faut toujours fermer la fenêtre avant d'essayer d'en ouvrir une nouvelle. Pour se déplacer dans la fenêtre on utilise la fonction suivante :

```
Graphics.moveto : int -> int -> unit
```

et pour tracer une ligne partant de la position courante on utilise la fonction suivante :

```
Graphics.lineto : int -> int -> unit
```

Pour changer la couleur on utilise :

```
Graphics.set_color : color -> unit
```

Il existe évidemment de nombreuses autres fonctions, qui vous simplifieront la vie (et ce TP), vous les trouverez dans la documentation de Caml :

<http://caml.inria.fr/pub/docs/manual-ocaml/>

## Exercice 1 : images et arbres

Une image est généralement représentée comme une matrice de couleurs. Cependant il existe d'autres représentations, plus compactes, des images. Nous nous intéressons ici à une représentation d'images sous formes d'arbres. Pour des raisons de simplicité nous considérons uniquement des images carrées, de côté  $2^n$ . De telles images peuvent être représentées de la manière suivante : une image unie est représentée par sa couleur et une image non unie est représentée par un arbre, de la manière proposée dans la figure 1.

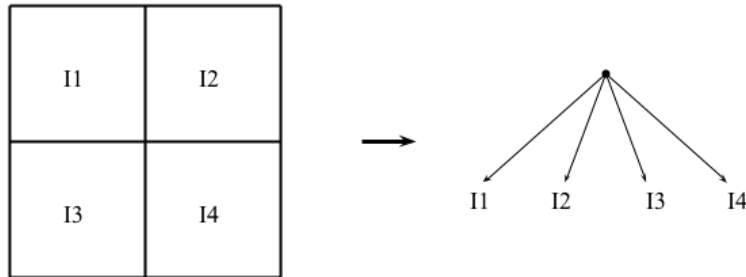


FIGURE 1 – Arbre d'une image

On a donc les types suivants :

```
type couleur = Blanc | Noir;;
```

```
type image_arb = Feuille of couleur | Noeud of arbre * arbre * arbre * arbre;;
```

Et pour les images vues comme des matrices :

```
type image_mat = couleur array array;;
```

**Question 2.** Écrivez une fonction qui affiche dans la fenêtre graphique une image représentée par un arbre. Écrivez une fonction qui effectue la rotation d'une image de 90 degrés dans le sens horaire. Écrivez une fonction qui inverse les couleurs d'une image. Écrivez une fonction qui transforme une image représentée par une matrice en une image représentée par un arbre (si possible la même image).

## Exercice 2 : flocon de Von Koch

La courbe de Von Koch peut être obtenue à partir d'un segment en utilisant l'algorithme suivant :

1. diviser le segment en trois parties égales,
2. remplacer la partie du milieu par un triangle équilatéral ayant pour base cette partie,
3. supprimer la base du triangle,
4. retourner à l'étape 1) pour chaque segment de la courbe obtenue.

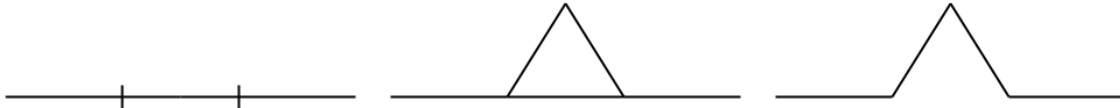


FIGURE 2 – Construction de la courbe de Von Koch

La figure 2 représente la première itération de cet algorithme.

On obtient un flocon de Von Koch en construisant une courbe de Von Koch sur chaque côté d'un triangle équilatéral (triangles orientés vers l'extérieur) ou sur chaque côté d'un hexagone (triangles orientés vers l'intérieur).

**Question 3.** Écrivez une fonction construisant une courbe de Von Koch à partir d'un segment. Écrivez une fonction construisant un flocon de Von Koch.

### Exercice 3 : jeu de la vie

Le jeu de la vie est un automate cellulaire (un modèle où chaque état conduit à l'état suivant à partir de règles pré-établies) imaginé par John H. Conway dans les années 70. Le jeu se déroule sur une grille à deux dimensions dont chaque case (ou cellule) peut prendre deux états : « vivante » ou « morte ». L'état d'une cellule au temps (discret)  $t + 1$  est déterminé par l'état de ses huit voisines (et le sien) au temps  $t$  selon des règles simples :

1. une cellule « morte » au temps  $t$  dont exactement 3 des voisines sont « vivantes » au même temps  $t$  devient « vivante » au temps  $t + 1$ ,
2. une cellule « vivante » au temps  $t$  dont 2 ou 3 des voisines sont « vivantes » au même temps  $t$  reste « vivante » au temps  $t + 1$ ,
3. une cellule ne vérifiant aucune des deux premières règles au temps  $t$  est « morte » au temps  $t + 1$ .

La figure 3 présente un exemple d'évolution selon ces règles (les cellules contenant un point noir sont « vivantes »).

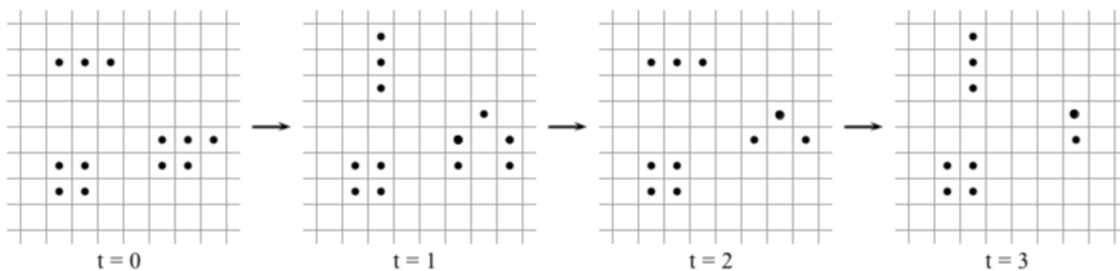


FIGURE 3 – Exemple d'évolution selon les règles du jeu de la vie.



**Question 4.** Écrivez une fonction qui affiche une boîte simple en fonction des coordonnées de son centre. Écrivez une fonction calculant la largeur d'une boîte (non simple) et une fonction calculant la hauteur d'une boîte (non simple). Écrivez une fonction qui affiche une boîte (non nécessairement simple) en fonction des coordonnées de son centre.

## Exercice 5 : enveloppe convexe

En dimension 2, l'enveloppe convexe d'un ensemble  $E$  de points est le plus petit polygone convexe contenant tous les points de  $E$  (un exemple figure 5). Il existe de nombreux algorithmes pour calculer ce polygone. Dans cet exercice nous nous intéressons à un de ces algorithmes, appelé marche de Jarvis. Ce n'est pas le plus efficace mais il a l'avantage d'être très simple. Pour simplifier on considérera dans un premier temps qu'il n'existe pas 3 points alignés dans l'ensemble  $E$ . Le principe de cet algorithme est le suivant : on part d'un point  $p_0 \in E$  dont on sait qu'il appartient à l'enveloppe convexe et, par la suite, on choisit le point  $p_{i+1} \in E$  à partir du point  $p_i$  de telle sorte que tous les autres points soient « à droite » de la droite allant de  $p_i$  à  $p_{i+1}$ .

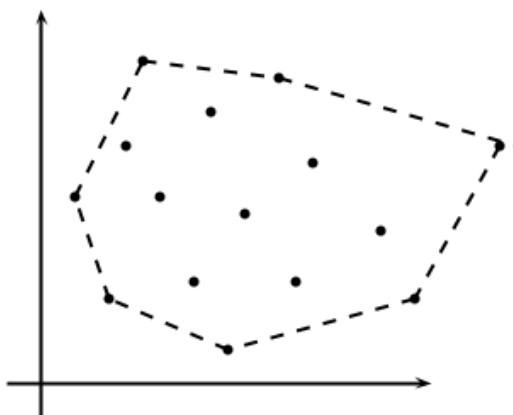


FIGURE 5 – Enveloppe convexe d'un ensemble de points

**Question 5.** Écrivez une fonction qui calcule (à l'aide l'algorithme proposé) puis affiche l'enveloppe convexe d'un ensemble de points (dans lequel il n'y a pas 3 points alignés).

**Question 6.** Modifiez votre fonction pour pouvoir traiter n'importe quel ensemble de points.