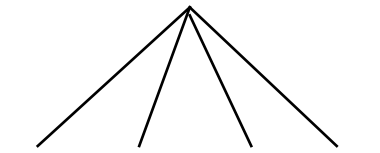
Travaux Pratiques de OCaml - TP 3

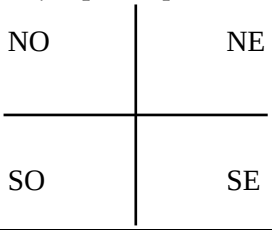
On définit inductivement la structure d'arbre quaternaire, quadtree, comme suit :

1- Les atomes B et N sont des quadtree.
2- Si A, B, C, D sont des quadtree alors le terme Nd(A,B,C,D) est un quadtree.
3- Tout quadtree est obtenu par un nombre fini d'applications des règles ci-dessus.

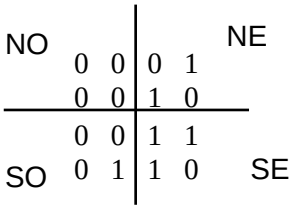
1) Définir en OCaml le type qd implémentant les quadtree en prenant les 3 constructeurs B,N et Nd.

On considère des images numériques de $2^n * 2^n$ points, $n \in N$. Chaque point est blanc (représenté par 0 dans les exemples ci-dessous et par le constructeur B en OCaml) ou noir (représenté par 1 dans les exemples ci-dessous et par le constructeur N en OCaml). Une telle image peut être représentée par un arbre quaternaire plein (toutes les feuilles sont à la même hauteur h). Une image (non réduite à un point) est composée de 4 images (NO, NE, SO et SE) qui se composent elles-mêmes d'images et ceci jusqu'aux points.

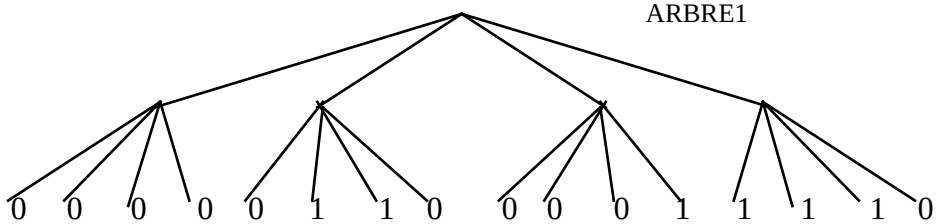




Exemple1. image1



ARBRE1

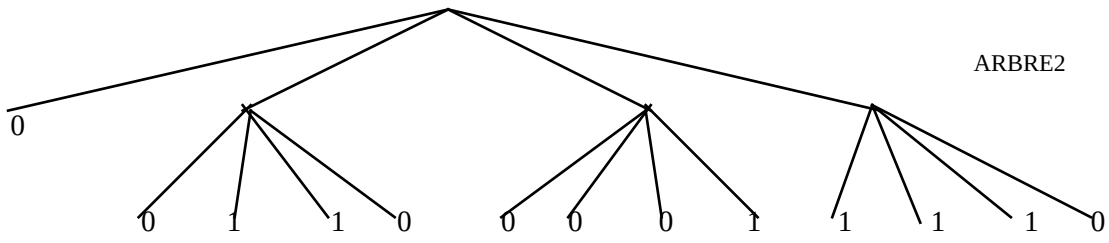


ARBRE1 représente l'arbre quaternaire plein pour l'image1.

#let arbre1 = Nd(Nd(B,B,B,B) , Nd(B,N,N,B) , Nd(B,B,B,N) , Nd(N,N,N,B)) ;;

Cette première représentation est la base d'une représentation plus compacte dite minimale obtenue par minimisation d'un arbre quaternaire : une image dont tous les points sont de même couleur est réduite à une feuille de cette couleur. Ainsi, une image uniformément blanche (resp. noire) est représentée en OCaml par B (resp. N).

ARBRE2



ARBRE2 représente l'arbre quate naire minimal pour l'image1.

#let arbre2 = Nd(B , Nd(B,N, N, B) , Nd(B,B,B,N) , Nd(N,N,N, B)) ;;

Remarque : on trouvera dans le fichier /users/linfg/b odeveix/TpCaml/TP3/affquad.ml la fonction affquad d'affichage de quadtree, de type qd * int -> unit, où int est la hauteur de l'arbre, ainsi que des exemples ...

2) Ecrire la fonction nbnoirs qui compte le nombre de points noirs d'une image (arbre plein).

3) Construction d'un arbre minimal.

a- Ecrire la fonction minquad qui appliquée à 4 arbres quaternaires a1, a2, a3 et a4 retourne a1 si a1, a2, a3 et a4 sont des atomes identiques et Nd(a1,a2,a3,a4) sinon.

b- Ecrire la fonction minimiser qui construit l'arbre minimal à partir d'un arbre quaternaire :

```
# minimiser Nd(Nd(B,B,B,B) , Nd(B,N, N, B) , Nd(B,B,B,N) , Nd(N,N,N, B)) ; ;
-: quadtree = Nd(B, Nd(B,N, N, B) , Nd(B,B,B,N) , Nd(N,N,N, B))
```

DANS CE QUI SUIT ON MANIPULERA EXCLUSIVEMENT DES ARBRES MINIMAUX.

4) Ecrire les fonctions :

- inverse qui inverse les points noirs et blancs d'une image.
- miroir_horizontal qui contruit le miroir d'une image par rapport à l'axe médian horizontal.
- miroir_vertical qui contruit le miroir d'une image par rapport à l'axe médian vertical.

5) Superposition d'images.

La superposition de deux images (de même dimension) est une image dans laquelle un point est noir si et seulement si le point correspondant est noir dans l'une ou l'autre des deux images. Ecrire la fonction superposition qui construit un arbre minimal

ARBRE3 représente l'arbre quaternaire minimal pour l'image2.

```
#let arbre3 = Nd(Nd(B , Nd(B,N,B,B) , B , Nd(B,B, N, N)), N, Nd (Nd(N,B,N,N) , Nd (N, N, N, B), B, N),
                Nd(N,N,N, B)) ;;
```