

TD07 - COMPILATEUR DE QUADTREES

On s'intéresse ici à la représentation d'images par des quadrees, utilisés pour l'analyse, la compression et la synthèse d'images. Une image est dite *simple* si elle est de couleur uniforme. Étant donné un médium graphique (écran, imprimante, etc.) carré, qui mesure 1024×1024 pixels, on peut représenter une image par une structure d'arbre, appelé *quadtree*, dans laquelle chaque nœud a exactement 0 ou 4 fils. L'idée de base est la suivante : si une image associée à un nœud n'est pas simple, on la découpe en 4 morceaux de même taille et les fils du nœud correspondant sont les *quadrees* associés aux 4 sous-images. Si au contraire l'image est simple, alors elle est caractérisée par une information unique, sa couleur ; dans ce cas, le nœud porte cette information et n'a pas de fils. Par exemple, si on parcourt les 4 quarts d'une image en décrivant un "Z", et si on suppose que les dix régions en lesquelles on a découpé le carré de la figure 1 sont simples, de couleurs **r**, **g** et **b**, alors cette figure peut être représentée par le *quadtree* de la figure 2. Un quadtree peut alors être représenté sous la forme d'une expression parenthésée qui décrit la structure de l'arbre correspondant. Le quadtree de la figure 2 est représenté sous la forme suivante : **(rbg((bbrg)grb))**

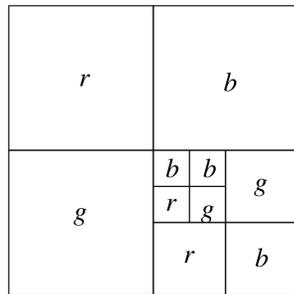


Figure 1

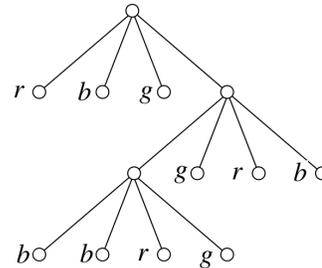


Figure 2

- (1) Écrivez une grammaire G des quadrees considérant l'alphabet des couleurs $\{r, g, b\}$.
- (2) Dessinez l'arbre de dérivation du quadtree $((bbr)(gbrg)rg)(bbrb)rb$
- (3) Écrivez une grammaire attribuée pour compter le nombre s de sous-carrés simples qui composent un quadtree. La valeur de cet attribut à la racine de l'arbre de dérivation pour l'exemple de la figure 1 est $s = 10$.

On dispose par ailleurs d'un périphérique graphique qui reconnaît l'unique instruction élémentaire

RECT $x_0 y_0 x_1 y_1 c$

dans laquelle x_0, y_0, x_1 et y_1 sont des nombres entiers. L'exécution de cette instruction produit le remplissage du rectangle $R = \{(x, y) \mid x_0 \leq x < x_1 \quad y_0 \leq y < y_1\}$ avec la couleur c . Un programme pour notre périphérique, appelé *programme-machine*, est une suite de telles instructions.

- (4) En vous aidant de la figure 1, écrivez le programme-machine du *quadtree* **(rbg((bbrg)grb))**.
- (5) Écrivez une grammaire attribuée pour les attributs x_0, y_0 et t contenant les coordonnées x et y du coin supérieur gauche du carré et la taille t de ses côtés.
- (6) On souhaite connaître le nombre de pixels de chacune des trois couleurs. Pour cela on crée les attributs n_R, n_G et n_B qui indiquent le nombre de pixels de chaque couleur. Écrivez une grammaire attribuée pour les attributs n_R, n_G et n_B .
- (7) Construire la table SLR pour cette grammaire.
- (8) Écrivez la spécification de la grammaire et du lexique au format **Sablecc**

- (9) Ecrivez un visiteur de l'arbre de dérivation produit par `Sablecc` qui produit le langage-machine du périphérique ci-dessus à partir d'un quadtree. Notez que $x_1 = x_0 + \text{taille}$ et $y_1 = y_0 + \text{taille}$.