Le modèle **Arbres.mod**

Pour TeXgraph 1.95

10 février 2011

Table des matières

1 Présentation

Le modèle *Arbres* permet une création simplifiée d'arbres dans TeXgraph par l'intermédiaire de la souris, sans néanmoins posséder toute la puissance de l'extension *pst-tree* de la suite *pstricks*.

Ce modèle est une extension du modèle *Mouse.mod*, qui permet d'ajouter à l'arbre des éléments graphiques à la souris (courbes fléchées, cadres ...)

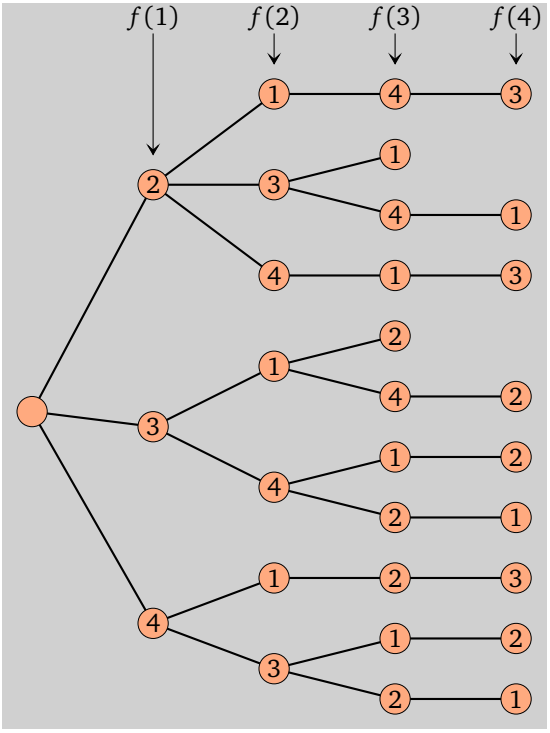

FIGURE 1 – Les 9 dérangements des entiers de 1 à 4

Après le chargement du modèle, seule la racine de l'arbre apparaît, celle-ci correspond à la variable globale $N0$, et son indice (0) apparaît sous la racine. On peut ajouter un descendant (un node) par un clic gauche sur la racine, ce node va avoir un indice k, il correspondra à la variable globale Nk, l'utilisateur doit alors fournir le texte associé au node (celui-ci sera enregistré dans la macro *node<k>*, par exemple *node0*, *node1* ...) puis éventuellement le label qui sera affiché le long de la branche reliant le node à son parent (ce label est enregistré dans la macro *lab<k>*, par exemple *lab0*, *lab1* ...).

2 Le menu

Celui-ci est constitué de tous les boutons du menu du modèle *Mouse.mod*, auquels viennent s'ajouter quatre boutons (en bas de la liste), qui sont :

1. **Aide sur l'arbre** : celui-ci sert uniquement à afficher un message rappelant comment ajouter, modifier ou supprimer un node à la souris, et propose d'ouvrir le fichier *Arbes.pdf* en demandant éventuellement le lecteur de pdf et son chemin d'accès.
2. **Déplacer un node** : cette option permet à l'utilisateur de déplacer un node (et sa descendance) vers la descendance d'un autre node (dit node hôte). Si ce dernier est dans la descendance du premier, alors la commande est sans effet.
3. **Orienter l'arbre** : cette option permet à l'utilisateur de choisir l'orientation de l'arbre, celle-ci est définie par deux directions : Dir1 et Dir2, la première indique le sens des branches de l'arbre, la seconde est orthogonale à la première et indique l'ordre d'affichage des nodes d'un même niveau. Huit choix sont proposés :
   - *RD* : vers la droite et en bas (c'est le cas dans la figure 1).
   - *RU* : vers la droite et en haut,
   - *LD* : vers la gauche et en bas,
   - *LU* : vers la gauche et en haut,
   - *DR* : vers le bas et vers la droite,
   - *DL* : vers le bas et vers la gauche,
   - *UR* : vers le haut et vers la droite,
   - *UL* : vers le haut et vers la gauche.
4. **Ajuster la fenêtre** : ce bouton est une bascule qui permet d'activer ou désactiver (valeur 0 ou 1 de la variable globale *Ajuster*) l'ajustement automatique de la fenêtre à l'arbre. Par défaut, la valeur est 1 (ajustement automatique). Lorsque l'arbre est terminé, il peut parfois être intéressant de désactiver l'ajustement automatique pour agrandir la fenêtre et ajouter d'autres éléments graphiques à la souris.

3 Les paramètres d'un arbre

Les différents paramètres apparaissent dans la liste des variables globales. On modifie ces paramètres en éditant la variable globale correspondante par un double clic. Ces paramètres sont :

- **armpos** : valeur entre 0 et 1 qui détermine la position du coude lorsque les branches sont dessinées avec *linetype*=1 ou 2 (ligne brisée),
- **labeloriente** : valeur 0 ou 1 qui indique si les labels, le long des branches, sont orientés parallèlement aux branches (valeur 1), ou sont horizontaux (valeur 0).
- **treelabelpos** : valeur entre 0 et 1 qui détermine la position du label le long de la branche (à partir du coude lorsque *linetype*=1 ou 2). Lorsque cette valeur est inférieure à 0.5, le label est calé à gauche, lorsqu'elle vaut 0.5, le label est centré, sinon il est calé à droite,
- **treelabelsep** : valeur positive qui détermine la distance (en cm) entre la branche et son label (lorsqu'il y en a un),
- **nodeep** : valeur positive qui détermine la taille (en cm) des nodes de l'arbre,
- **treeheight** : valeur positive qui détermine la hauteur (en cm) d'un niveau de l'arbre,
- **treesep** : valeur positive qui détermine la distance (en cm) entre deux arbres voisins,
- **linetype** : valeur entière valant : 0, 1, 2, 3, ou 4 qui détermine le type de raccordement entre deux nodes,
  - *linetype=0* : segment de droite,
  - *linetype=1* : segment à un coude, la position du coude par rapport au parent est déterminée par le paramètre *armpos*,
  - *linetype=2* : segment à 2 coudes, la position du coude par rapport au parent est déterminée par le paramètre *armpos*,
  - *linetype=3* : courbe de Bézier,
  - *linetype=4* : segment de droite en forme de tube, celui-ci est en fait un segment dessiné avec les attributs *Width**3 et Color, puis le segment est redessiné avec les attributs *Width* et *FillColor*.

4 Compléments

4.1 Modifier l'apparence des branches

Les attributs des branches : *Color*, *FillColor* et *Width*, sont les attributs de l'élément graphique *arbre*. Il suffit donc de modifier les attributs de cet élément pour changer ces paramètres.

On peut changer aussi la forme des branches en modifiant la macro *drawAline(depart, direction, indice1, indice2)* qui relie le node d'indice %3 (qui est à l'affixe %1) au node d'indice %4 (le vecteur direction est donné par %2).

4.2 Modifier l'apparence des labels

Les attributs des labels : *Color* et *LabelSize*, sont les attributs de l'élément graphique *arbre*. Il suffit donc de modifier les attributs de cet élément pour changer ces paramètres.

On peut changer aussi l'aspect des labels en modifiant la macro *drawAlabel(depart, direction, label, indice1, indice2)* qui dessine le label le long de la branche reliant le parent d'affixe %1 (et d'indice %4) au node d'indice %5, le vecteur direction est %2 et le texte du label est %3.

4.3 Modifier l'apparence des nodes

Les attributs des nodes : *Color*, *FillStyle*, *FillColor*, *LabelStyle* et *LabelSize*, sont les attributs de l'élément graphique *styleNodes*. Il suffit donc de modifier les attributs de cet élément pour changer ces paramètres.

On peut changer aussi la forme des nodes en modifiant la macro *drawAnode(indice)* qui dessine le node d'indice %1. Par défaut, chaque node est un label dessiné avec *LabelStyle=framed*, *LineStyle=noline*, *FillStyle=full* et *Fillcolor=white*. Si par exemple, on veut un cercle autour de chaque node, alors on peut modifier la macro *drawAnode* comme suit :

```
[{on dessine le node indice %1, les attributs
 sont ceux de l'élément graphique styleNodes}
 Eval( ["Cercle(N",%1, ",", nodeep, ")" ]),
 Eval( ["Label( N", %1, ", \node", %1, ")" ])
]
```

Il faudra ensuite modifier les attributs de l'élément graphique *styleNodes* pour mettre *LineStyle=solid*.
En faisant un test sur l'indice, on peut ne modifier que certains nodes.

4.4 Permuter deux descendants d'un même node

Supposons par exemple que l'on veuille permuter les descendants d'indices 3 et 4 du node 1. On édite la variable globale *arbre1* par un double clic, celle-ci est une liste commençant par l'indice 1 (c'est le parent), suivi des indices des fils ; il suffit de permuter les indices 3 et 4 dans cette liste et valider.

4.5 Un deuxième exemple

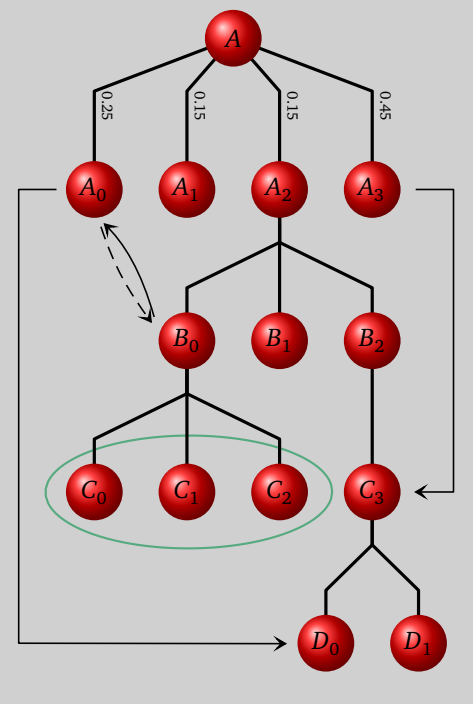

FIGURE 2 – Exemple de modifications des macros

Cet exemple est une exportation au format *pgf*. Pour obtenir ces jolies sphéres rouges à reflet blanc, nous avons déclaré un type de shading [1] dans le document LaTeX avant d'inclure l'image (*Arbre2.pgf*).

Pour cela nous avons dans la macro `UserBsave`, écrit le texte suivant :

```
if ExportMode=pgf then
 WriteFile("\pgfdeclareradialshading{ballshading}{\pgfpoint{-10bp}{10bp}}
{color(0bp)=(red!15!white);color(9bp)=(red!75!white);
color(18bp)=(red!70!black);color(25bp)=(red!50!black);color(50bp)=(black)}")
fi
```

Cette macro est appelée par la macro `Bsave`, elle sera donc exécutée à chaque export. La commande `WriteFile` écrit la chaine passée en argument dans le fichier d'exportation.

TeXgraph ne sait pas faire ce type de shading, aussi avons-nous modifié la macro *drawAnode* de la manière suivante :

```
[{drawAnode(indice): on dessine le node indice %1,
  les attributs sont ceux de l'élément graphique styleNodes}
if Export And (ExportMode=pgf) then
$z:= Eval(["N",%1]) ,
Special("\pgfpathcircle{\pgfxy\[\coord(z)\]}{\[nodeep\]cm}
\pgfshadepath{ballshading}{0}\pgfusepath{}"),
 else Eval( ["Cercle(N",%1, ",", nodeep, ")" ])
fi,
Eval( ["Label( N", %1, ", \node", %1, ")" ])
]
```

1. Cet exemple de shading est extrait de la documentation du package pfg : *pgfmanual.pdf*.

Au moment de l'export la macro *Bsave* (qui est modifiée dans *Arbres.mac*) est exécutée, elle met la variable *Export* à la valeur 1 et recalcule l'arbre et les indices. Lors du recalcul de l'arbre la macro *drawAnode* est exécutée pour chaque node, comme la variable *Export* vaut 1, et si on exporte en pgf, alors on écrit un label spécial, c'est à dire qu'il est écrit tel quel dans le fichier exporté, à la différence près que tout ce qui est entre deux balises `\[` et `\]` est interprété par TeXgraph.. Plus précisément :

- On calcule z qui est l'affixe du node en coordonnées TEX, supposons par exemple que $z = 3 + 4.5i$ et que la variable *nodeep* vaut 0.375,
- on ecrit dans le fichier les deux lignes :

```
\pgfpathcircle{\pgfxy(3,4.5)}{0.375cm}
\pgfshadepath{ballshading}{0}\pgfusepath{}
```

Lorsque la variable *Export* est nulle, c'est un cercle centré sur le node et de rayon 0.375 qui est dessiné (à l'écran ou dans les autres exports que pgf).

D'autre part, on peut remarquer que les branches issues de la racine ne sont pas du même type que les autres (1 seul coude), pour cela nous avons modifié les macros *drawAline* et *drawAlabel* pour faire un test sur l'indice du node parent.

Pour terminer nous avons ajouté quelques flèches et une ellipse à la souris.