Le fichier drawFunction.mod (0.1)

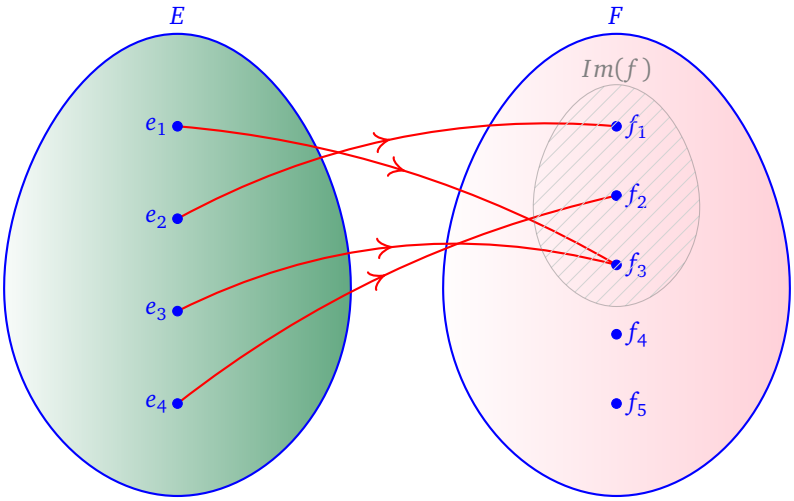
Pour TeXgraph 1.97

5 janvier 2013

Résumé

Description des macros de drawFunction.mac et drawFunction.mod.

Table des matières



1 Introduction

Le fichier modèle *drawFunction.mod* charge en mémoire le modèle *draw2d.mod* puis les macros du fichier *drawFunction.mac*, ces dernières permettent le dessin d'un diagramme sagittal entre deux ensembles finis. Lorsque ce modèle est chargé dans l'interface graphique, on voit apparaître un exemple. Il suffit d'éditer l'élément graphique pour pouvoir adapter son code.

2 Utilisation

Celle-ci se fait en plusieurs étapes :

1. Modification éventuelles des options globales qui sont :
   - topsep := < nombre > pour définir la distance entre le haut de l'ensemble et le premier élément (1 par défaut).

- bottomsep := < nombre > pour définir la distance entre le bas de l'ensemble et le dernier élément (1 par défaut).
- elemsep := < nombre > pour définir la distance entre deux éléments de l'ensemble (0.75 par défaut).
- setsep := < nombre > pour définir la distance entre les deux ensembles (1 par défaut).

2. Modifications des options par défaut pour les flèches avec la macro :

   arrowDefaultOptions([options])

   Ces options sont :
   - arrowprm := < nombre > pour la courbure de la flèche. Celle-ci peut être négative et elle vaut 1 par défaut.
   - arrowpos := < nombre dans [0; 1] > définit la position de la flèche le long de la courbe de Bézier (0.5 par défaut ce qui représente le milieu).
   - arrowtype := < nombre > définit le type de flèche, on peut utiliser tous les types définis dans le modèle *draw2d.mod*. La valeur par défaut est *Carrow*.

3. Définition des données et options concernant les deux ensembles et le graphe avec la macro :

   initData(arg1, arg2, arg3, arg4, arg5, arg6, arg7 [, arg8, arg9, ...])

   où :
   - arg1 est le nom de l'ensemble de départ, par exemple `"E"`.
   - arg2 est la liste des éléments de l'ensemble de départ, ceux-ci peuvent être des nombres ou des chaînes, par exemple `["e_1","e_2","e_3"]`.
   - arg3 est la liste des options à appliquer à l'ensemble de départ, par exemple `[Color:=blue, Width:=8]`.
   - arg4 est le nom de l'ensemble d'arrivée, par exemple `"F"`.
   - arg5 est la liste des éléments de l'ensemble d'arrivée, ceux-ci peuvent être des nombres ou des chaînes, par exemple `[1,2,3]`.
   - arg6 est la liste des options à appliquer à l'ensemble d'arrivée, par exemple `[Color:=red, Width:=6]`.
   - arg7 est une liste constituée d'un nombre pair d'entier représentant le graphe, par exemple la liste `[1,2, 2,1, 3,3]` signifie que le premier élément de E (si E est l'ensemble de départ) est en relation avec le deuxième élément de F (si F est l'ensemble d'arrivée), le deuxième élément de E est en relation avec le premier élément de F et le troisième élément de E est en relation avec le troisième élément de F.
   - arg8 est une liste d'options s'appliquant à la première flèche du graphe, par exemple : `[LineStyle:=userdash, arrowpos:=0.6]`.
   - arg9 est une liste d'options s'appliquant à la deuxième flèche du graphe ... etc.

4. Dessin du graphe avec avec l'instruction :

   draw("graph", [options globales]

   Les options globales sont les attributs que l'on souhaite pour le dessin (couleur, épaisseur, ...), plus l'option graphsize := < taille > qui permet d'imposer une taille au graphique sous la forme d'un complexe *largeur+i*hauteur*, si la hauteur est omise elle est considérée comme égale à la largeur. Par défaut la valeur de ce paramètre est *Nil* ce qui signifie que le graphe aura sa taille « naturelle ».

3 Exemple

Voici l'élément graphique créé lors du chargement du modèle dans l'interface graphique de TeXgraph :

Exemple

```
\begin{texgraph}[name=drawFunction2, file]
Include "drawFunction.mod";
Graph image = [
 topsep :=1,        //espace élément-haut de l'ensemble
 bottomsep :=1,     //espace élément-bas de l'ensemble
 elemsep :=0.75,    //espace entre deux éléments
 setsep :=1,        //espace entre les deux ensembles
 dollar :=1,        //ajouter ou non des dollars autour des labels
 initData(
```

```
 "E",                    //nom ensemble départ
 ["e_1", "e_2", "e_3", "e_4"],  //liste des éléments de E
 [Color :=blue],              //options dessin E
 "F",                    //nom ensemble arrivée
 [1,2,3],                  //liste des éléments de F
 [Color :=red],               //options dessin F
 [1,2, 2,3, 3,1, 4,2],       //graphe ici f(e_1)=2, f(e_2)=3 f(e_3)=1 f(e_4)=2
 [LineStyle :=userdash],        //options flèche 1
 [Color :=gray],              //options flèche 2 ... etc
),
arrowDefaultOptions(
 [arrowprm :=1,            //paramètres courbure flèche
  arrowpos :=0.5,          //position flèche
  arrowtype :=Carrow       //type de flèche (modèle draw2d)
 ]),
draw("graph",                //dessin du graphe
    [Width :=8, graphsize :=Nil]   //options globales
  )
];
\end{texgraph}
```

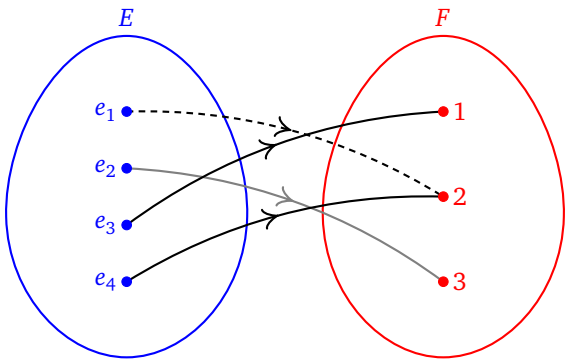


FIGURE 1: *Fonction surjective mais non injective.*

Voici l'exemple donné en préambule :

Exemple

```
\begin{texgraph}[name=drawFunction3, file]
Include "drawFunction.mod";
Graph image = [
 FrenchBabel :=1,
 initData(
 "E",                    //nom ensemble départ
 ["e_1", "e_2", "e_3", "e_4"],  //liste des éléments de E
 [Color :=blue,FillStyle :=gradient,FillColorB :=seagreen],  //options dessin E
 "F",                    //nom ensemble arrivée
 ["f_1","f_2","f_3","f_4","f_5"],  //liste des éléments de F
 [Color :=blue,FillStyle :=gradient,FillColorB :=pink],   //options dessin F
 [1,3, 2,1, 3,3, 4,2],       //graphe
 [arrowprm :=0.75],[],[arrowprm :=1.25],[arrowprm :=0.75]
 ),
 draw("graph",                //dessin du graphe
    [Width :=8,Color :=red,graphsize :=12+8*i]   //options globales
   ),
 x :=(poselem(2,1)+poselem(2,3))/2,
 setB("Im(f)",x,[labelsep :=0.2,Color :=gray,FillStyle :=bdiag,
          StrokeOpacity :=0.5,FillColor :=lightgray,scale :=0.3])
];
\end{texgraph}
```

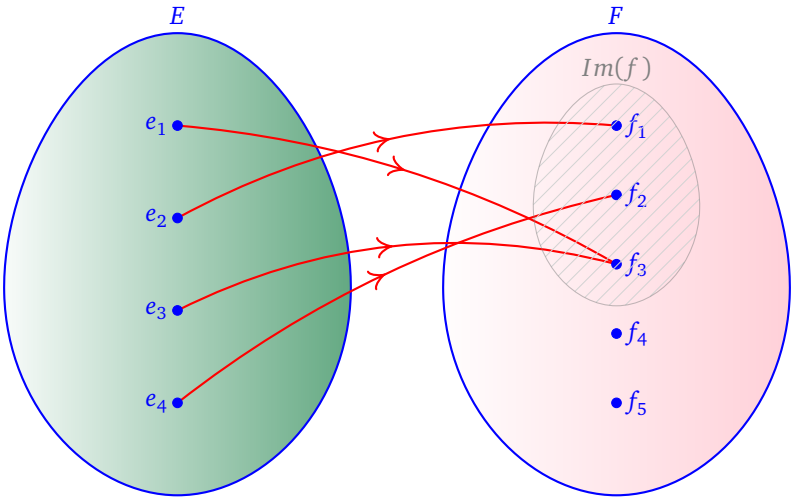


FIGURE 2: *Fonction ni injective ni surjective*