

# Fonctions

## 1 Le début

Dans un exercice, j'ai vu la définition suivante :

Soit la fonction  $f : \mathbb{R} \longrightarrow \mathbb{R}, x \longmapsto x + 1$ .

On obtient cet affichage en écrivant :

```
f\,:\,\mathbb{R} \longrightarrow \mathbb{R}, $x \longmapsto x+1$
```

On doit pouvoir faire mieux!

## 2 Question de goût

Doit-on écrire  $\mathbb{R}$  ou **R** ?

Certains disent que l'écriture  $\mathbb{R}$  (à double barre) a été créée pour ce que l'on écrit à la main, mais que pour un texte qui doit être imprimé, c'est le gras qui doit être utilisé. Personnellement, j'emploie donc **N, Z, D, Q, R** ou **C**.

Naturellement, on pourra créer une commande `\R` pour **R** :

```
\newcommand{\R}{\textbf{R}}
```

ou, si on préfère la double barre  $\mathbb{R}$  :

```
\newcommand{\R}{\mathbb{R}}
```

## 3 Première approche

### 3.1 Mise en place

En utilisant un tableau de type `array`, on arrive facilement à :

Soit la fonction  $f : \mathbf{R} \longrightarrow \mathbf{R}$   
 $x \longmapsto x + 1$

En voici le code :

```
Soit la fonction
$\begin{array}[t]{l r c l}
f\,,: & \mathbf{R} & \longrightarrow & \mathbf{R} \\
& x & \longmapsto & x+1
\end{array}$
```

Ne pas oublier de mettre `[t]` (pour `top`) comme option dans la définition de `array` sinon on obtient :

Soit la fonction  $f : \mathbf{R} \longrightarrow \mathbf{R}$   
 $x \longmapsto x + 1$

Plutôt moche!

### 3.2 Automatisation

L'idéal est de définir une fonction qui fait le boulot.

Combien de paramètres ?

Un pour le nom de la fonction, un pour l'ensemble de départ, un pour l'ensemble d'arrivée, un pour le nom de la variable, et un pour l'image ; soit un total de cinq.

Voici le code de cette fonction appelée `\fonct` :

```
\newcommand{\fonct}[5]{
\begin{array}[t]{l r c l}
#1\,,: & #2 & \longrightarrow & #3\ \\
& & & & #4 & \longmapsto & & #5 \\
\end{array}}
```

On utilise alors cette fonction ainsi :  $\$ \backslash \text{fonct} \{ f \} \{ \mathbb{R} \} \{ \mathbb{R} \} \{ x \} \{ x+1 \} \$$

## 4 Un peu mieux

### 4.1 Mise en place

Si on trouve les flèches `\longrightarrow` et `\longmapsto` un peu courtes, on peut en dessiner d'autres. Toujours avec un tableau de type `array`, on définit des nœuds avec `\Rnode`, et on les relie avec `\ncline` ; cela donne :

$$\begin{array}{ccc}
 f : \mathbf{R} & \longrightarrow & \mathbf{R} \\
 x & \longmapsto & x + 1
 \end{array}$$

dont le code est :

```
\[\psset{nodesep=5pt,arrowsize=2pt 2,linewidth=0.5pt}
\begin{array}[t]{@{} l r @{\hspace*{2cm}} l}
f\,,: & \Rnode{dep}{\mathbb{R}} & \Rnode{arr}{\mathbb{R}} \ncline{->}{dep}{arr} \\
& \Rnode{xx}{x} & \Rnode{fx}{x+1} \ncline{|->}{xx}{fx} \\
\end{array}\]
```

On définit deux nœuds appelés `dep` et `arr` qui représentent les ensembles de départ et d'arrivée, puis on dessine une flèche qui va de `dep` vers `arr`.

On définit deux autres nœuds appelés `xx` et `fx` qui représentent la variable et son image ; on les relie par une flèche.

La distance entre les extrémités des flèches et les nœuds est réglée par `nodesep`, et la longueur des flèches dépend du `@{\hspace*}` qui sert de séparateur entre deux colonnes du tableau.

Toutes les explications sont détaillées dans le mode d'emploi de `pst-tree`.

### 4.2 Automatisation

Là aussi on peut souhaiter définir une fonction qui fait le boulot.

Combien de paramètres ?

Un pour la longueur des flèches, un pour le nom de la fonction, un pour l'ensemble de départ, un pour l'ensemble d'arrivée, un pour le nom de la variable, et un pour l'image ; soit un total de six.

Voici le code de cette fonction appelée `\fonx` :

```
\newcommand{\fonx}[6]{
\psset{nodesep=5pt,arrowsize=2pt 2,linewidth=0.5pt}
\begin{array}[t]{@{} l r @{\hspace*{#1}} l}
#2\,: & \Rnode{dep}{#3} & \Rnode{arr}{#4} \ncline{->}{dep}{arr}\\
& & \Rnode{xx}{#5} & \Rnode{fx}{#6} \ncline{|->}{xx}{fx}
\end{array}}
```

On utilise alors cette fonction ainsi :  $\$ \backslash \text{fonx} \{ 2 \text{cm} \} \{ f \} \{ \mathbb{R} \} \{ \mathbb{R} \} \{ x \} \{ x+1 \} \$$

On peut aussi ne pas vouloir changer souvent la taille des flèches si on a trouvé une taille adéquate (2 cm par exemple); on va donc mettre le paramètre de largeur en option avec une valeur par défaut de 2cm; c'est possible car la longueur des flèches est le premier paramètre de la fonction.

On doit donc modifier la première ligne, ce qui donne pour définition de la commande qu'on appelle `fonc` :

```
\newcommand{\fonc}[6][2cm]{
\psset{nodesep=5pt,arrowsize=2pt 2,linewidth=0.5pt}
\begin{array}[t]{@{} l r @{\hspace*{#1}} l}
#2\,: & \Rnode{dep}{#3} & \Rnode{arr}{#4} \ncline{->}{dep}{arr}\\
& & \Rnode{xx}{#5} & \Rnode{fx}{#6} \ncline{|->}{xx}{fx}
\end{array}}
```

On utilise alors cette fonction ainsi :  $\$ \backslash \text{fonc} \{ f \} \{ \mathbb{R} \} \{ \mathbb{R} \} \{ x \} \{ x+1 \} \$$  pour avoir des flèches de taille 2 cm.

Et si on veut modifier la taille des flèches, on entrera la nouvelle longueur comme premier paramètre mais entre crochets.

Par exemple, « Soit  $\$ \backslash \text{fonc} [ 4 \text{cm} ] \{ g \} \{ \mathbb{R} \} [ 0^{\sim}; \sim + \infty ] [ t ] \{ \text{e}, \} \{ t \} \$$  » donne :

$$\text{Soit } g : \begin{array}{ccc} \mathbf{R} & \longrightarrow & ]0; +\infty[ \\ t & \longmapsto & e^t \end{array}$$