

# CHAPITRE 11

---

## Dessin

---

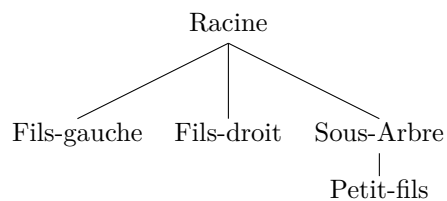
On pourrait rédiger plus d'un livre sur le dessin avec  $\text{\LaTeX}$ . Dans ce chapitre, on va rapidement passer sur les deux packages prédominants que sont `PSTricks` et `TikZ/PGF`. On verra également différents packages permettant de faire des dessins plus ciblés comme des arbres grammaticaux pour les linguistes, des molécules pour les chimistes, des circuits pour les électriciens ...

### 11.1 Arbres

Pour dessiner des arbres, on peut utiliser le package spécialisé `qtree`. Ce dernier permet de facilement dessiner des arbres dont on donne la description avec la commande `\Tree`. On décrit un arbre par un simple mot pour une feuille ou par une liste de nœuds séparés par des espaces et mis entre crochets pour un sous-arbre, la racine étant précédée d'un point.

```
\Tree [.Racine Fils-gauche Fils-droit [.Sous-Arbre Petit-fils ] ]
```

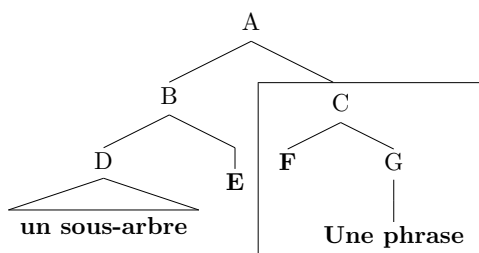
Il faut bien veiller à ce que chaque crochet soit précédé d'une espace.



Les arbres sont automatiquement centrés. Pour annuler ce comportement, il suffit de spécifier l'option `nocenter` au moment de charger le package. On peut modifier le style en redéfinissant les commandes vides suivantes :

- `qtreeinithook` est appelée au début de chaque arbre
- `qtreefinalhook` est appelée à la fin de chaque arbre
- `qleafhook` est appelée pour chaque feuille
- et `qlabelhook` est appelée pour chaque nœud interne

Enfin, on peut obtenir tout un sous-arbre sous forme d'un triangle avec la commande `\qroof`. On peut également encadrer un sous-arbre avec la commande `\qframesubtree`. Remarquez aussi qu'on n'est pas obligé d'avoir un texte pour chaque nœud interne. Voici un exemple plus élaboré :



Remarquez que `\qframesubtree` ne fonctionnera pas si vous avez le package `babel` qui est chargé.

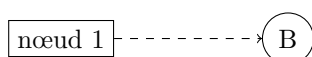
**Code**
qtree

```

\newcommand{\qleafhook}{\bf}
\Tree [.A
  [.B
    \qroof{un sous-arbre}.D
    [E ]
  ]
  [.C
    [.F ]
    [.G [{Une phrase} ] ]
  ] !{\qframesubtree}
]
```

## 11.2 TikZ/PGF

Un package de dessin très connu et fonctionnant avec pdfTeX est le package TikZ/PGF. Nous n'allons pas le détailler ici car cela nécessiterait un livre complet, mais on va simplement regarder un ou deux exemples. L'idée de base est de créer des nœuds qu'on va ensuite relier par des arêtes. On insère une nouvelle figure avec l'environnement `tikzpicture`. Voyons un premier exemple.



**Code**

tikz,pgf

```

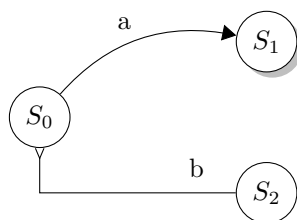
\begin{tikzpicture}
  \node[draw,shape=rectangle] (A) at (0,0) {nœud 1};
  \node[draw,shape=circle] (B) at (3,0) {B};
  \draw (A) edge[dashed,->] (B);
\end{tikzpicture>

```

La commande `\node` permet d'ajouter un nouveau nœud. On définit d'abord des options, ensuite un nom entre parenthèses suivi de `at` et de la coordonnée du nœud et enfin, on place le texte qui doit figurer dans le nœud entre accolades.

La commande `\draw` permet de faire un dessin. Dans notre cas, on dessine une arête entre le nœud A et le nœud B. Pour cela, on utilise `edge` qui permet notamment de définir des options.

Il existe de nombreuses extensions pour TikZ/PGF. L'exemple suivant montre l'utilisation de deux extensions. La première permet d'avoir des styles de flèches différents et la seconde permet d'avoir des ombres. Vous pourrez également voir sur l'exemple suivant qu'on peut définir des options par défaut et qu'on peut ajouter des étiquettes sur les arêtes.



**Code**

tikz,pgf

```

\begin{tikzpicture}[shape=circle,auto]
  \node[draw] (A) at (0,0) { $S_0$ };
  \node[draw,fill=white,drop shadow] (B) at (3,1) { $S_1$ };
  \node[draw] (C) at (3,-1) { $S_2$ };
  \draw (A) edge[-triangle 60,bend left] node {a} (B);
  \draw[open reversed 45-] (A) |- node[pos=0.9] {b} (C);
\end{tikzpicture>

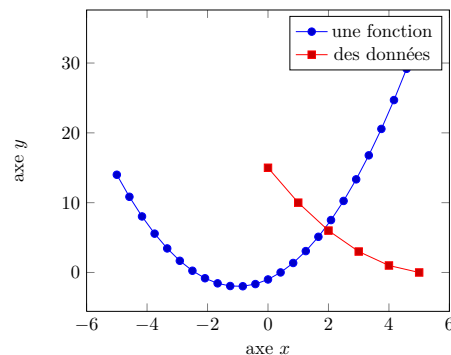
```

Il faut préalablement importer les deux extensions avec la commande `\usetikzlibrary` dans le préambule du document :

```
\usetikzlibrary{arrows,shadows}
```

### 11.2.1 Dessiner des données

On peut dessiner des données en utilisant le package `pgfplots` qui se base sur le package TikZ/PGF.



**Code**

pgfplots

```

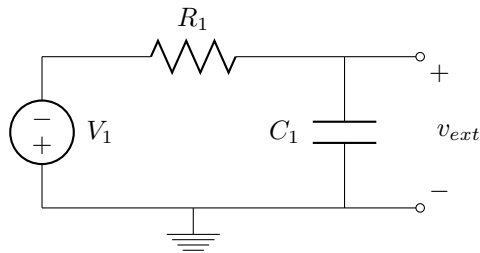
\begin{tikzpicture}[scale=0.7]
  \begin{axis}[xlabel=axe  $x$ ,ylabel=axe  $y$ ]
    \addplot {x^2 + 2*x - 1};
    \addlegendentry{une fonction}
    \addplot coordinates {
      (0,15)
      (1,10)
      (2,6)
      (3,3)
      (4,1)
      (5,0)
    };
    \addlegendentry{des données}
  \end{axis}
\end{tikzpicture>

```

On utilise l'environnement `axis` pour définir un nouveau graphe. Ensuite, on ajoute une courbe grâce à la commande `\addplot`. On peut soit définir une équation, soit donner une série de points. Enfin, on peut obtenir une légende avec `\addlegendentry`. Il existe de nombreuses autres possibilités offertes par ce package et nous vous invitons à consulter sa documentation détaillée.

### 11.2.2 Dessiner des circuits électriques

Le package `circuitikz` permet de dessiner des circuits électriques facilement. Il se base, comme son nom le suggère, sur TikZ/PGF. Il y a essentiellement des monopoles et des dipôles. Chacun de ces éléments est représenté par un nœud. On va utiliser `node` pour les monopoles et `to` pour les dipôles.



**Code**
circuitikz

```

\begin{circuitikz}[american]
  \draw (-2,1) to[voltage source,l=$V_1$] (-2,-1);
  \draw (-2,1) to[R,l=$R_1$] (2,1);
  \draw (2,1) to[short] (3,1);
  \draw (2,1) to[C,l=$C_1$] (2,-1);
  \draw (-2,-1) to[short] (3,-1);
  \draw (0,-1) node[ground] {};
  \draw (3,1) to[open,o-o,v^=$v_{ext}$] (3,-1);
\end{circuitikz}

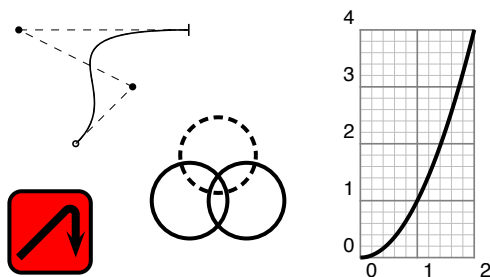
```

N'hésitez pas à consulter la documentation officielle pour avoir la liste des tous les composants qu'il est possible de dessiner.

### 11.3 PSTricks

PSTricks est un package de dessin qui permet d'exploiter les possibilités de PostScript avec L<sup>A</sup>T<sub>E</sub>X. Ce package n'est pas compatible avec pdfT<sub>E</sub>X et vous devrez dès lors passer par un document en PostScript à convertir en un PDF.

Les dessins se font grâce à des commandes à qui il faut passer, en plus d'options et de paramètres, des coordonnées sous la forme (x,y).



Code
pstricks, pst-all

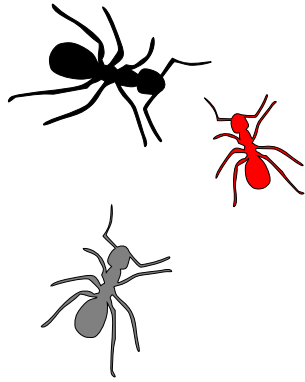
```

\pspicture(0,0)(8,5)
  \psframe[linewidth=2pt,framearc=0.3,fillstyle=solid,
           fillcolor=red](-0.2,-0.3)(1.3,1.2)
  \psline[linewidth=4pt,linearc=0.1]{->}(0,0)(1,1)(1,0)
  \pscicle[linewidth=2pt](3,1){0.7}
  \pscicle[linewidth=2pt,linestyle=dashed](3.5,1.8){0.7}
  \pscicle[linewidth=2pt](4,1){0.7}
  \psbezier[showpoints=true]{o-|}(1,2)(2,3)(0,4)(3,4)
  \psgrid[origin={6,0},gridcolor=gray,
          subgridcolor=lightgray](0,0)(0,0)(2,4)
  \psplot[origin={6,0},plotpoints=100,
          linewidth=2pt]{0}{2}{x x mul}
\endpspicture

```

Il existe bien d'autres commandes ainsi que de nombreux autres packages qui ont été construits sur base de PSTricks. On peut notamment dessiner des montages pour des expériences de chimie, faire des dessins 3D ...

Voici par exemple une utilisation du package `pst-fun` qui permet notamment d'insérer des fourmis avec la commande `\pstAnt`.

**Code****pstricks, pst-all, pst-fun**

```
\pspicture(0,0)(8,5)
  \rput(1,1){\psscalebox{0.5}{\psAnt}}
  \rput{45}(3,3){\psscalebox{0.4}{\psAnt[fillcolor=red]}}
  \rput{-80}(1,4){\psscalebox{0.6}{\psAnt[fillcolor=black]}}
\endpspicture
```

