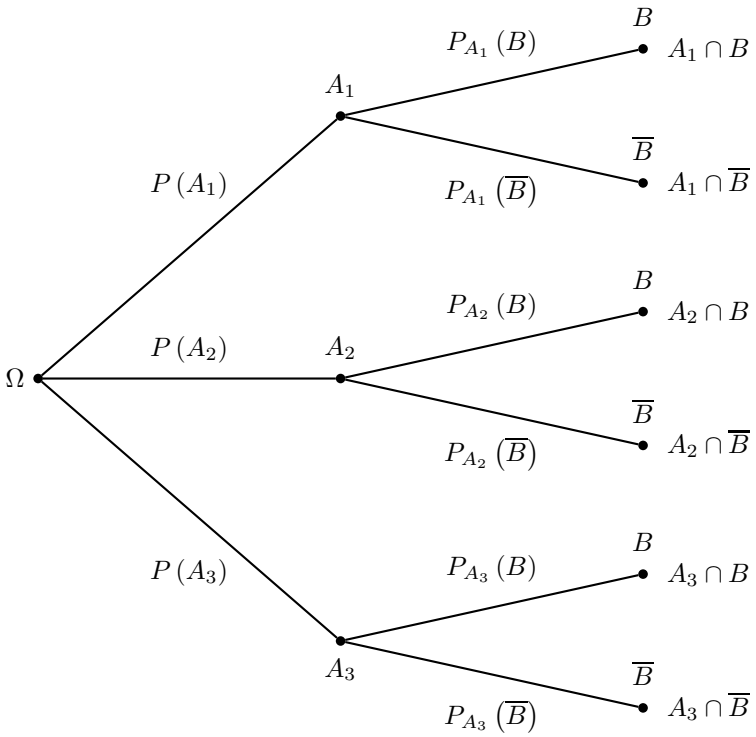
Chronique 5

Arbres

5.1 Objectif

Dans cette chronique, on va voir comment créer des arbres et comment les modifier.
On aboutira à un arbre assez compliqué qui devrait couvrir quelques besoins :

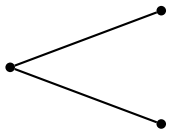


Il y a des dizaines de façons de présenter des arbres, je ne vous en montrerai que quelques-unes.

5.2 Premiers pas

Il existe plusieurs extensions qui permettent de construire des arbres ; j'utilise `pst-tree` que je charge dans le préambule par `\usepackage{pst-tree}` (`pst` pour `PsTricks`).
On trouve la documentation de ce package (en anglais) sur l'immanquable site `CTAN` à l'adresse :
`http://ctan.mines-albi.fr/graphics/pstricks/contrib/pst-tree/pst-tree-doc.pdf`
Il y a même le document source `tex`.

On va commencer par un tout petit bout d'arbre :



dont voici le code (très simple) :

```
\psset{treemode=R, nodesep=0mm, levelsep=20mm, treesep=15mm}
\pstree{\Tdot}{\Tdot \Tdot}
```

En première ligne on définit les principaux paramètres de l'arbre :

- `treemode=R` désigne le sens vers lequel va être construit l'arbre : `R` pour `right`, `L` pour `left`, `U` pour `up` et `D` (valeur par défaut) pour `down` ;
- `nodesep=0mm` désigne la distance entre le nœud et le début de la branche qui va y être attachée ;
- `levelsep=20mm` désigne la distance horizontale entre deux nœuds quand l'arbre est dirigé vers la droite, et c'est aussi la longueur d'une branche horizontale ;
- `treesep=15mm` désigne la distance entre deux nœuds situés en bout d'arbre (ou à peu près !).

En deuxième ligne, on rencontre l'instruction `\pstree` qui permet tout !
Cette instruction nécessite deux paramètres ; le premier désigne ce que l'on va mettre à la racine de l'arbre, le deuxième paramètre correspond aux successeurs :

- `{\Tdot}` en premier paramètre désigne un point comme racine ;
- `{\Tdot \Tdot}` en second paramètre désigne les deux branches des successeurs.

On peut mettre d'autres choses que `\Tdot` :

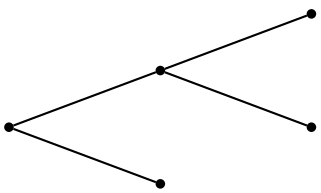
- `\Tf` pour dessiner un petit carré ;
- `\TC` pour dessiner un cercle dont on fixe le rayon par exemple à 2 points en rajoutant l'option `radius=2pt` dans la ligne `\psset` ;
- `\TC*` pour dessiner un disque (dont le rayon est lui aussi géré par `radius`) ;
- `\Tp` pour ne rien dessiner du tout (mais il faut mettre quelque chose quand même !).

5.3 Arbres plus fournis

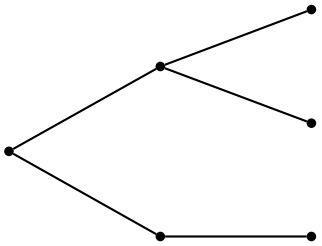
Pour faire évoluer le petit arbre précédent, il suffit de remplacer un point `\Tdot` par un sous-arbre `\pstree{}{}`.

Voici quelques variations sur le même principe.

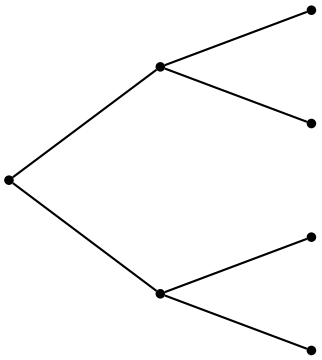
```
\psset{treemode=R,nodesep=0mm,%
          levelsep=20mm,treesep=15mm}
          % paramètres

\pstree{\Tdot}  % racine
{
    \pstree     % sous-arbre supérieur
        {\Tdot}       % racine
        {\Tdot \Tdot} % successeurs
    \Tdot % branche inférieure
}
```



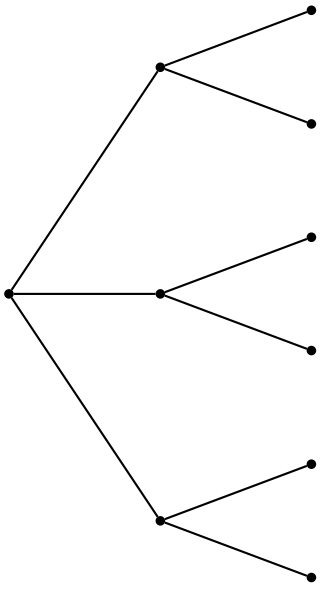
```
\psset{treemode=R,nodesep=0mm,%
          levelsep=20mm,treesep=15mm}
\pstree{\Tdot} % racine
{
    \pstree{\Tdot}{\Tdot \Tdot}
       % sous-arbre du haut
    \pstree{\Tdot}{\Tdot}
       % sous-arbre du bas
}
```



```
\psset{treemode=R,nodesep=0mm,%
          levelsep=20mm,treesep=15mm}
\pstree{\Tdot}
{
    \pstree{\Tdot}{\Tdot \Tdot}
    \pstree{\Tdot}{\Tdot \Tdot}
}
```



```
\psset{treemode=R,nodesep=0mm,%
          levelsep=20mm,treesep=15mm}
\pstree{\Tdot}
{
    \pstree{\Tdot}{\Tdot \Tdot}
    \pstree{\Tdot}{\Tdot \Tdot}
    \pstree{\Tdot}{\Tdot \Tdot}
}
```



On s'approche de l'objectif de cette chronique : on a construit un arbre à trois branches qui se séparent chacune en deux branches.
La différence entre ce dernier arbre et celui de la page 23 est la longueur des branches qui a été allongée dans le premier arbre pour permettre l'écriture des légendes ; cela a été obtenu par `levelsep=40mm` au lieu de `levelsep=20mm`.

Il ne reste plus qu'à nommer les nœuds et mettre des poids sur les branches, comme dans tout arbre pondéré qui se respecte !

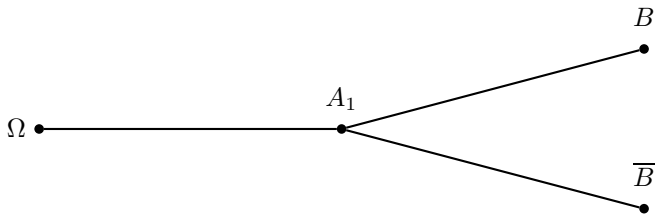
5.4 Les nœuds

Pour écrire le nom d'un nœud, c'est très simple ; il faut l'attacher au moyen d'un tilde (∼) au point que l'on trace ; le tilde s'obtient par `Alt Gr 2`.
On définit la position du nom au moyen de la variable `\tnpos` à laquelle on peut donner la valeur `a` pour `above`, `b` pour `below`, `l` pour `left` ou `r` (par défaut) pour `right` ; ainsi pour la racine on écrira `\Tdot~[tnpos=l]{$\Omega$}`, pour A_1 on écrira `\Tdot~[tnpos=a]{$A_1$}`, etc.

La branche supérieure de l'arbre de la page 23 sera donc construite ainsi :

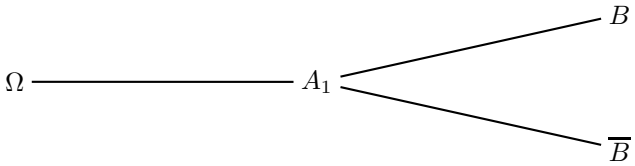
```
\psset{treemode=R, nodesep=0mm, levelsep=40mm, treesep=15mm}
\pstree{\Tdot~[tnpos=l]{$\Omega$}}
{
    \pstree
        {\Tdot~[tnpos=a]{$A_1$}}
        {
        \Tdot~[tnpos=a]{$B$}
        \Tdot~[tnpos=a]{$\overline{B}$}
        }
}
```

et donnera :



5.5 Autre version

Je sens bien que certains d'entre vous préfèreraient l'arbre précédent représenté ainsi :



Il y a le nom du nœud à la place du point ; on va donc remplacer `\Tdot~` par `\Tr` et laisser un peu de place autour du nom en modifiant la variable `nodesep` dans `\pset` :

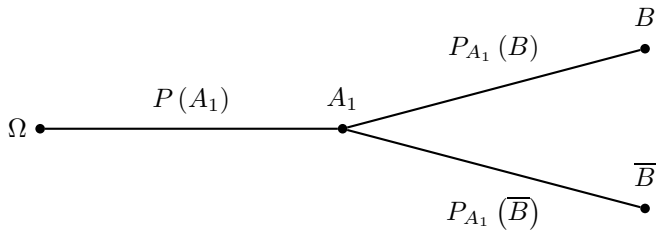
```
\psset{treemode=R, nodesep=1mm, levelsep=40mm, treesep=15mm}
\pstree{\Tr{$\Omega$}}
{
    \pstree
        {\Tr{$A_1$}}
        {
        \Tr{$B$}
        \Tr{$\overline{B}$}
        }
}
```

5.6 Les branches

Pour mettre des légendes sur les branches de l'arbre, on utilise les instructions `\taput` (avec `a` pour `above`), `\tbput` (avec `b` pour `below`) ou encore `\tlput` et `\trput` (respectivement `left` et `right`) quand l'arbre est dirigé vers le haut ou vers le bas.
Voici la version de l'arbre précédent avec les probabilités sur les branches :

```
\psset{treemode=R, nodesep=0mm,levelsep=40mm,treesep=15mm}
\pstree{\Tdot~[tnpos=l]{$\Omega$}}
   {\pstree
        {\Tdot~[tnpos=a]{$A_1$}\taput{$P\(A_1\)$}}
        {
        \Tdot~[tnpos=a]{$B$}\taput{$P_{A_1}\(B\)$}
        \Tdot~[tnpos=a]{$\overline{B}$}\tbput{$P_{A_1}\(\overline{B}\)$}
        }
   }
```

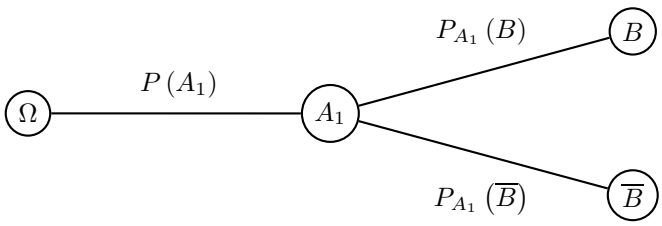
Ce qui donne comme résultat :



5.7 Nouvelle version

Peut-être voulez-vous une autre forme d'arbre ?
En voici une avec les noms des nœuds entourés par des cercles qui s'adaptent à la taille des noms ; il faut pour cela utiliser `\Tcircle` :

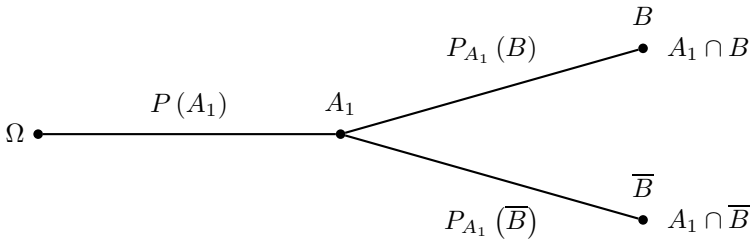
```
\psset{treemode=R, nodesep=0mm,levelsep=40mm,treesep=15mm}
\pstree{\Tcircle{$\Omega$}}
{
    \pstree
        {\Tcircle{$A_1$}\taput{$P\(A_1\)$}}
        {\Tcircle{$B$}\taput{$P_{A_1}\(B\)$}
        \Tcircle{$\overline{B}$}\tbput{$P_{A_1}\(\overline{B}\)$}}
}
```



Pour avoir des cercles de même taille, il faut utiliser `\TCircle` (avec un `C` majuscule) à la place de `\Tcircle` et définir le rayon des cercles par une option du style `radius=20pt`.

5.8 La touche finale

Et comment écrire $A_1 \cap B$ et $A_1 \cap \overline{B}$ au bout de l'arbre ?



Il suffit de rajouter les séquences `{$A_1 \cap B$}` et `{$A_1 \cap \overline{B}$}` précédées d'un tilde au bon endroit :

```
\psset{treemode=R, nodesep=0mm,levelsep=40mm,treesep=15mm}
\pstree{\Tdot~[tnpos=l]{$\Omega$}}
{
    \pstree
        {\Tdot~[tnpos=a]{$A_1$}\taput{$P\(A_1\)$}}
        {
        \Tdot~[tnpos=a]{$B$}~{$A_1 \cap B$}\taput{$P_{A_1}\(B\)$}
        \Tdot~[tnpos=a]{$\overline{B}$}~{$A_1 \cap \overline B$}
            \tbput{$P_{A_1}\(\overline{B}\)$}
        }
}
```

Une remarque : la séquence `\Tdot~[tnpos=a]{$B$}~{$A_1 \cap B$}` ne donne pas exactement le même résultat que la séquence `\Tdot~{$A_1 \cap B$}~[tnpos=a]{$B$}`.
Ne me demandez pas pourquoi !

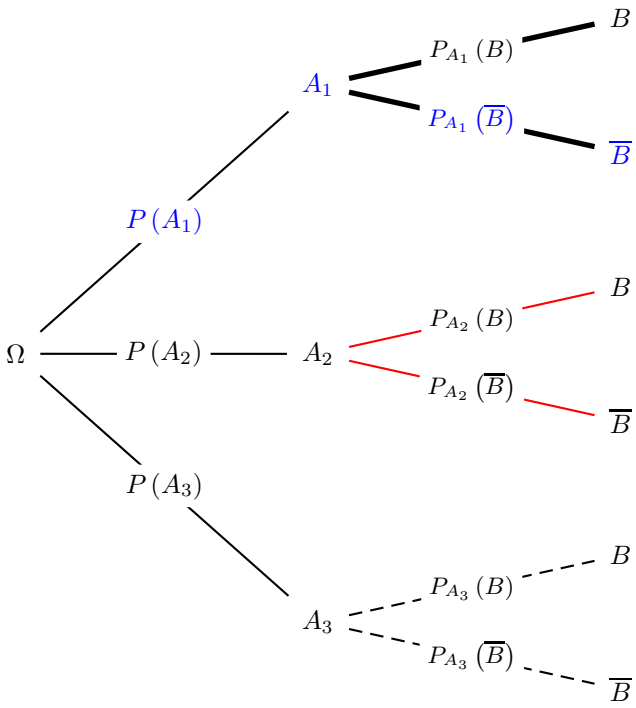
5.9 Le code

Voici le code complet de l'arbre de la page 23 :

```
\psset{treemode=R, nodesep=0mm, levelsep=40mm, treesep=10mm}
\pstree{\Tdot~[tnpos=l]{$\Omega$}}
{
    \pstree
        {\Tdot~[tnpos=a]{$A_1$}\taput{$P\(A_1\)$}}
        {\Tdot~[tnpos=a]{$B$}~{$A_1 \cap B$}\taput{$P_{A_1}\(B\)$}
        \Tdot~[tnpos=a]{$\overline{B}$}~{$A_1 \cap \overline B$}
            \tbput{$P_{A_1}\(\overline{B}\)$}}
    \pstree
        {\Tdot~[tnpos=a]{$A_2$}\taput{$P\(A_2\)$}}
        {\Tdot~[tnpos=a]{$B$}~{$A_2 \cap B$}\taput{$P_{A_2}\(B\)$}
        \Tdot~[tnpos=a]{$\overline{B}$}~{$A_2 \cap \overline B$}
            \tbput{$P_{A_2}\(\overline{B}\)$}}
    \pstree
        {\Tdot~[tnpos=b]{$A_3$}\tbput{$P\(A_3\)$}}
        {\Tdot~[tnpos=a]{$B$}~{$A_3 \cap B$}\taput{$P_{A_3}\(B\)$}
        \Tdot~[tnpos=a]{$\overline{B}$}~{$A_3 \cap \overline B$}
            \tbput{$P_{A_3}\(\overline{B}\)$}}
}
```

5.10 Variante

Au lieu de placer les légendes au dessus ou en dessous des branches, on peut les écrire sur les branches en utilisant l'instruction `\ncput*` à la place de `\taput` et `\tbput`.
Voici un exemple (à ne pas suivre!) :



et le code pour l'obtenir :

```
\psset{treemode=R, nodesep=2mm, levelsep=40mm, treesep=15mm}
\pstree{\Tr{$\Omega$}}
{
    \pstree[linewidth=2pt]
        {\Tr{\blue $A_1$} \ncput*{\blue $P\(A_1\)$}}
        {\Tr{$B$} \ncput*{\small $P_{A_1}\(B\)$}
        \Tr{\blue $\overline{B}$}
        \ncput*{\blue \small $P_{A_1}\(\overline{B}\)$}}
    \pstree[linecolor=red]
        {\Tr{$A_2$} \ncput*{$P\(A_2\)$}}
        {\Tr{$B$} \ncput*{\small $P_{A_2}\(B\)$}
        \Tr{$\overline{B}$} \ncput*{\small $P_{A_2}\(\overline{B}\)$}}
    \pstree[linestyle=dashed]
        {\Tr{$A_3$}\ncput*{$P\(A_3\)$}}
        {\Tr{$B$} \ncput*{\small $P_{A_3}\(B\)$}
        \Tr{$\overline{B}$} \ncput*{\small $P_{A_3}\(\overline{B}\)$}}
}
```

J'ai rajouté quelques options et un peu de couleur par-ci par-là pour vous donner des idées de modification.

5.11 Raccourcis

Il y a une variable appelée `shortput` qui permet d'utiliser des raccourcis pour les deux instructions `\taput` et `\tbput`.

L'instruction `\taput` place la légende au dessus de la branche, comme en exposant ; on pourra donc remplacer `\taput{$P\(A_1\)$}` par `^{$P\(A_1\)$}`.

De même comme l'instruction `\tbput` place la légende en dessous de la branche, comme en indice, on remplacera `\tbput{$P_{A_1}\(\overline{B}\)$}` par `_{$P_{A_1}\(\overline{B}\)$}`.

Il faut quand même activer la variable `shortput` et lui donner la valeur `tab`, ce que l'on fait dans `\psset` en entrant `shortput=tab`.

Voici le code qui donne exactement le diagramme du paragraphe 5.8, en utilisant les raccourcis :

```
\psset{treemode=D,nodesep=0mm,levelsep=40mm,treesep=15mm,shortput=tab}
\pstree{\Tdot~[tnpos=l]{$\Omega$}}
{
    \pstree
        {\Tdot~[tnpos=a]{$A_1$}^{$P\(A_1\)$}}
        {
        \Tdot~[tnpos=a]{$B$}~{$A_1 \cap B$}^{$P_{A_1}\(B\)$}
        \Tdot~[tnpos=a]{$\overline{B}$}~{$A_1 \cap \overline B$}
            _{$P_{A_1}\(\overline{B}\)$}
        }
}
```

Une interprétation (toute personnelle) de ce que peut vouloir dire `tab`.
Le `t` fait référence à `tree`.
Quant aux deux lettres de `ab`, elles veulent dire tout simplement `above` et `below` que l'on retrouve dans `\taput` et `\tbput`.

Il y a également une valeur de la variable `shortput` qui est `tablr` ; je ne suis pas loin de penser que dans ce cas le `l` signifie `left`, et le `r` signifie `right` ! ! !

Enfin dans une prochaine chronique sur les graphes, on verra que `shortput` peut prendre la valeur `nab`, avec `n` pour `node` (nœud), `a` pour `above` et `b` pour `below`.