

# tkz-bernoulli

Présenter, grâce à **tikz**,  
des arbres de Bernoulli.

▷ Commandes [fr] ou [en] ◁

Version 0.1.0 -- 9 septembre 2023

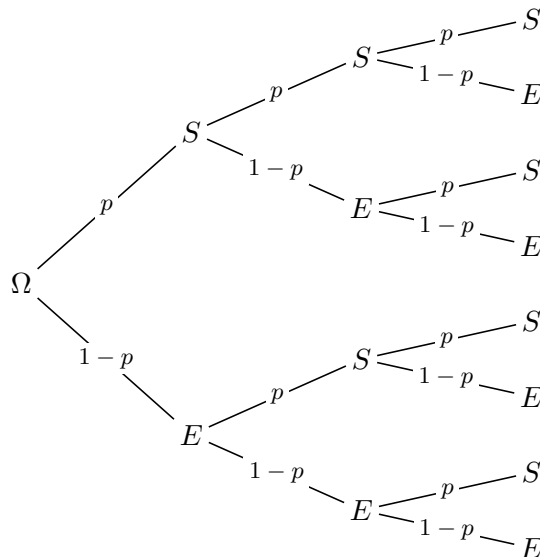
Cédric Pierquet

c pierquet -- at -- outlook . fr

<https://github.com/cpierquet/tkz-bernoulli>

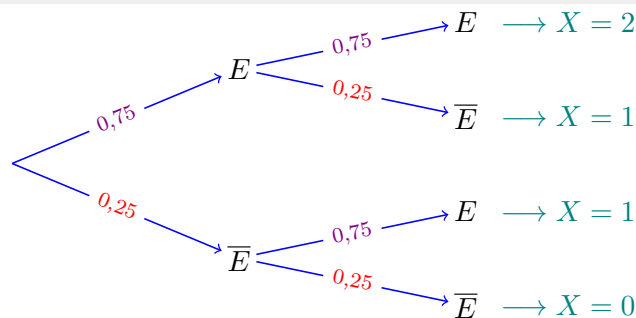
Présenter, avec personnalisations possibles, un arbre de Bernoulli.

```
\tkzSchemBernoulli*
```



```
\tkzSchemBernoulli*
```

```
[N=2,EspNiv=3,EspFeuil=1.25,NoticeProbas={\num{0.75}/\num{0.25}},Evts={\mathbb{E}/\overline{\mathbb{E}}}]
```



# Table des matières

<b>1</b>	<b>Le package tkz-bernoulli</b>	<b>2</b>
1.1	Introduction . . . . .	2
1.2	Chargement . . . . .	2
1.3	Commandes disponibles . . . . .	2
1.4	Styles par défaut . . . . .	3
<b>2</b>	<b>Les commandes</b>	<b>3</b>
2.1	Commande à insérer dans un environnement tikzpicture . . . . .	3
2.2	Commande autonome . . . . .	5
2.3	Modification avancée des styles . . . . .	8
<b>3</b>	<b>English commands</b>	<b>9</b>
3.1	Introduction . . . . .	9
3.2	Exemples . . . . .	9

---

## 1 Le package tkz-bernoulli

### 1.1 Introduction

L'idée du package `tkz-bernoulli` est de proposer des commandes pour représenter un schéma de Bernoulli, dans le cadre d'une loi binomiale par exemple, avec la possibilité de :

- personnaliser les dimensions et styles ;
- rajouter des éléments a posteriori, grâce aux nœuds créés.

### 1.2 Chargement

Le package se charge dans le préambule, via `\usepackage{tkz-bernoulli}`.

Les seuls packages chargés sont :

- `xstring`, `pgffor` et `simplekv` ;
- `xintexpr` et `xintbinhex` ;
- `tikz` avec la librairie `calc`.

```
\usepackage{tkz-bernoulli}
```

» `tkz-bernoulli` est compatible avec les compilations usuelles en `latex`, `pdflatex`, `lualatex` ou `xelatex`.

### 1.3 Commandes disponibles

Les commandes proposées par le package `tkz-bernoulli` sont :

```
%commande à insérer dans un environnement tikzpicture, pour rajouts éventuels
\begin{tikzpicture}
  \tkzSchemBernoulli
\end{tikzpicture}
```

```
%commande autonome
\tkzSchemBernoulli*
```

## 1.4 Styles par défaut

Le package propose des styles prédéfinis, pour :

- la racine et les nœuds ;
- les branches ;
- les probabilités.

Pour modifier, en *profondeur*, le style de l'arbre, il suffira de redéfinir les styles suivants (une commande est disponible pour remettre tous les styles par défaut) :

```
%style par défaut des branches
\tikzstyle{BernBranche} = [semithick]

%style par défaut du label de la racine, si affichée
\tikzstyle{BernRacine} = [inner sep=2pt]

%styles par défaut des noeuds relatifs à Succès/Échec
\tikzstyle{BernNoeudS} = [inner sep=2pt]
\tikzstyle{BernNoeudE} = [inner sep=2pt]

%styles par défaut des probas relatives à Succès/Échec
\tikzstyle{BernProbaS} = [fill=white,midway,font=\footnotesize,inner sep=1.5pt]
\tikzstyle{BernProbaE} = [fill=white,midway,font=\footnotesize,inner sep=1.5pt]

%style par défaut des valeurs prises par la v.a.
\tikzstyle{BernNotice} = [inner sep=2pt,text=teal,right=1em]

%commande de remise des styles par défaut
\tkzSchemBernStyleDefaut
```

## 2 Les commandes

### 2.1 Commande à insérer dans un environnement tikzpicture

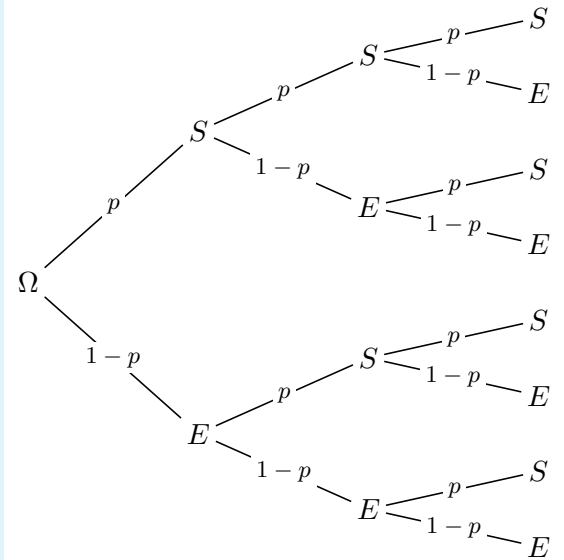
La commande dédiée pour insertion dans un environnement tikzpicture est \tkzSchemBernoulli :

```
%commande à insérer dans un environnement tikzpicture, pour rajouts éventuels
\begin{tikzpicture}
  \tkzSchemBernoulli[clés]
\end{tikzpicture}
```

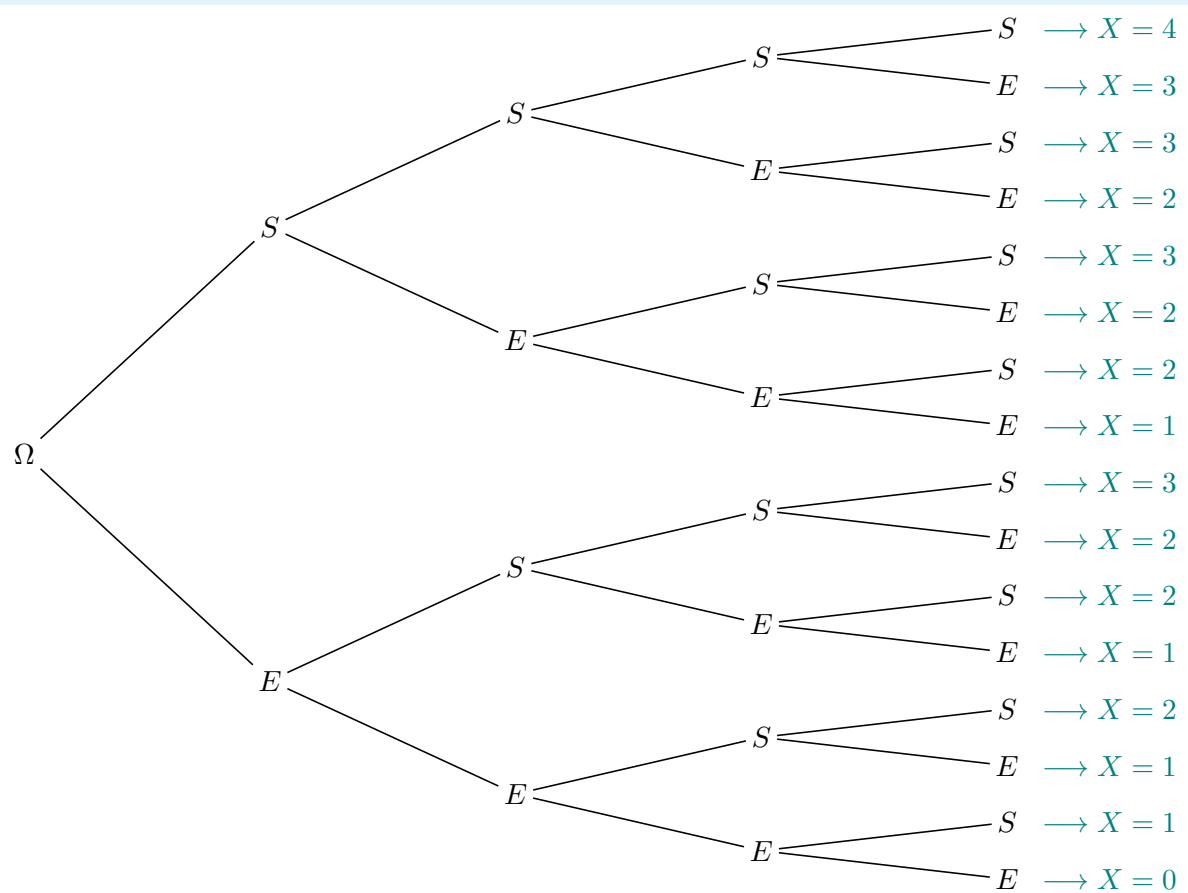
Concernant cette commande :

- les clés disponibles sont :
  - `EspNiv` := espace horizontal entre les niveaux (2.25 par défaut) ;
  - `EspFeuil` := espace vertical entre les éléments du dernier niveau (1 par défaut) ;
  - `Evts` := nom des évènements Succès/Échec (`$$/$E$` par défaut) ;
  - `Probas` := probabilités (`$p$/$1-p$` par défaut) ;
  - `AffProbas` := booléen pour afficher les probabilités (`true` par défaut) ;
  - `Racine` := nom qui apparaît pour la racine (`$\Omega$` par défaut, ou `false` pour désactiver) ;
  - `Aide` := booléen pour afficher les noms des nœuds créés (`false` par défaut) ;
  - `Notice` := booléen pour afficher les valeurs prises par la v.a. (`false` par défaut) ;
  - `Var` := nom de la v.a. pour la notice (`X` par défaut) ;
  - `N` := paramètre  $n$  du schéma de Bernoulli (3 par défaut).

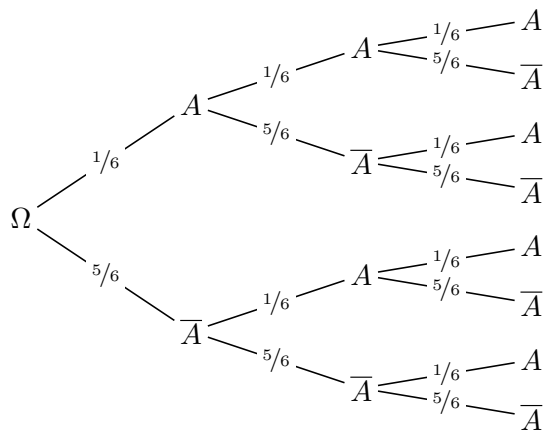
```
\begin{tikzpicture}
  \tkzSchemBernoulli
\end{tikzpicture}
```



```
\begin{tikzpicture}
  \tkzSchemBernoulli[Notice,AffProbas=false,EspNiv=3.25,EspFeuil=0.75,N=4]
\end{tikzpicture}
```



```
%\usepackage{nicefrac}
\begin{tikzpicture}
  \tkzSchemBernoulli[%
    Evts={ $A$ / $\overline{A}$ },%
    EspFeuil=0.75,%
    Probas={ $\frac{1}{6}$ / $\frac{5}{6}$ }]
\end{tikzpicture}
```



## 2.2 Commande autonome

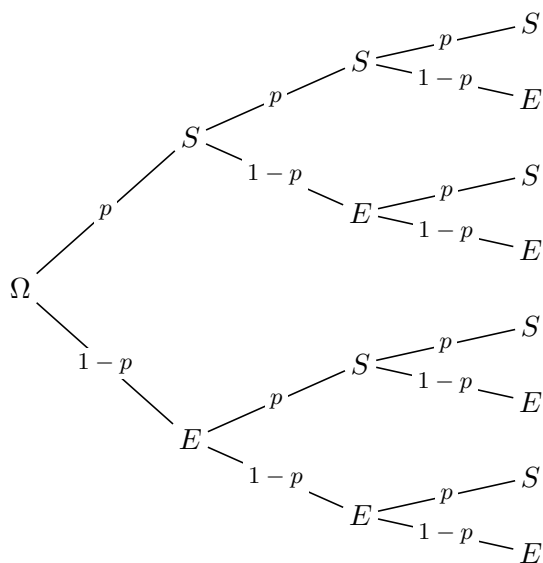
La commande dédiée pour insertion autonome est `\tkzSchemBernoulli*` :

```
%commande autonome
\tkzSchemBernoulli*[clés]<options tikz>
```

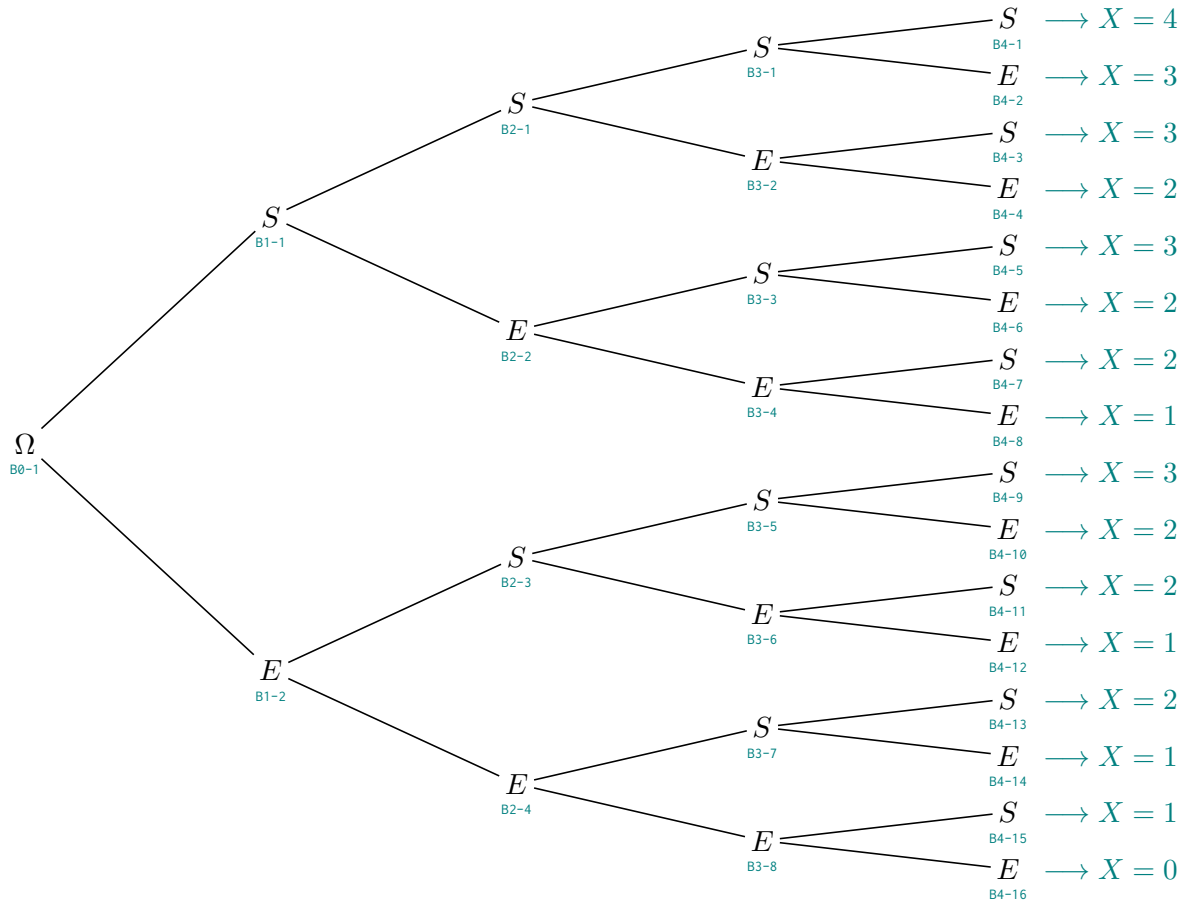
Concernant cette commande :

- l'environnement `tikzpicture` est automatiquement créé ;
- les `clés` sont les mêmes que pour la commande non étoilée ;
- des `<options tikz>`, optionnels, peuvent être passées à l'environnement `tikzpicture`.

```
\tkzSchemBernoulli*
```



\tkzSchemBernoulli\*[Notice,AffProbas=false,EspNiv=3.25,EspFeuil=0.75,N=4,Aide]



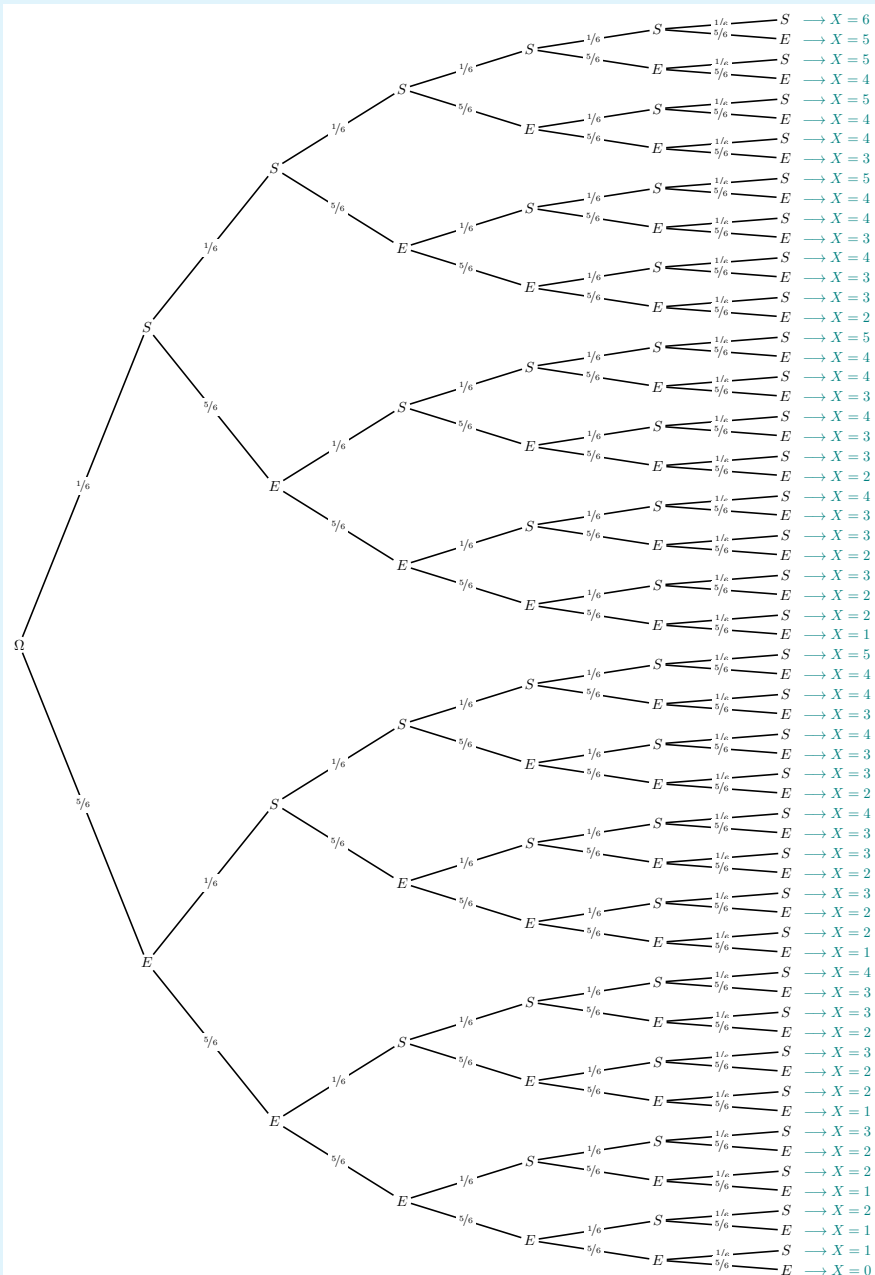
```
%\usepackage{nicefrac}
```

```
\tkzSchemBernoulli*[%
```

```
N=6,EspFeuil=0.35,Notice,%
```

```
Probas={ $\frac{1}{6}$ }{ $\frac{5}{6}$ }{ $\frac{1}{6}$ }{ $\frac{5}{6}$ }]
```

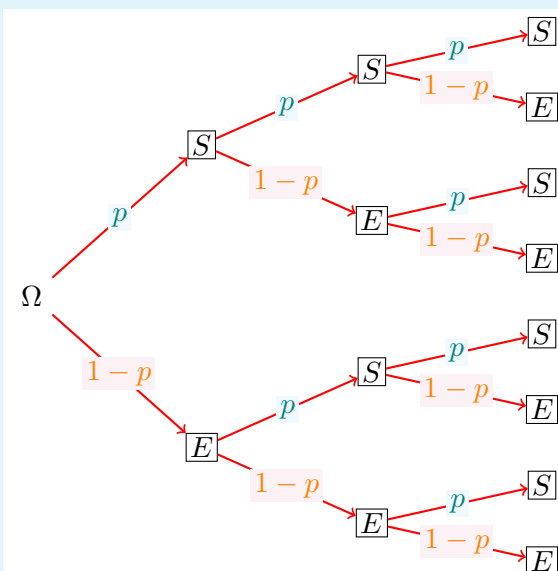
```
<scale=0.75,every node/.style={scale=0.5}>
```



## 2.3 Mofication avancée des styles

Les clés relatives aux commandes précédentes permettent de modifier l'aspect *global* de l'arbre, mais les styles particuliers des éléments peuvent également être modifiés, comme indiqué au début de cette documentation.

```
\tikzstyle{BernBranche} = [thick,red,->]
\tikzstyle{BernRacine} = []
\tikzstyle{BernNoeudS} = [draw,rectangle,inner sep=1.5pt]
\tikzstyle{BernNoeudE} = [draw,rectangle,inner sep=1.5pt]
\tikzstyle{BernProbaS} = [text=teal,midway,fill=cyan!5,inner sep=1.5pt]
\tikzstyle{BernProbaE} = [text=orange,midway,fill=purple!5,inner sep=1.5pt]
\tikzSchemBernoulli*
```





## 3 English commands

### 3.1 Introduction

There's also english versions of the commands and keys :

```
%command in an environment tikzpicture
\begin{tikzpicture}
  \tkzBernoulliTree[keys]
\end{tikzpicture}
```

```
%stand-alone command
\tkzBernoulliTree*[keys]<tikz options>
```

Default styles are given by :

```
%style for edges
\tikzstyle{BernEdge} = [semithick]

%style for root (if displayed)
\tikzstyle{BernRoot} = [inner sep=2pt]

%styles for nodes Success/Failure
\tikzstyle{BernNodeS} = [inner sep=2pt]
\tikzstyle{BernNodeF} = [inner sep=2pt]

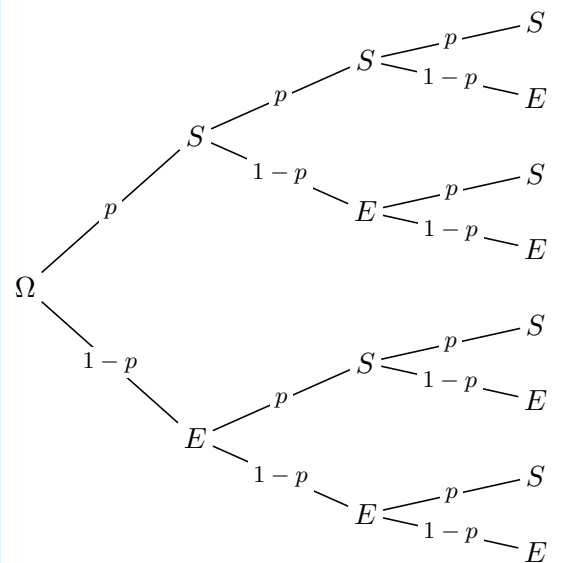
%styles for probas Success/Failure
\tikzstyle{BernProbS} = [fill=white,midway,font=\footnotesize,inner sep=2pt]
\tikzstyle{BernProbF} = [fill=white,midway,font=\footnotesize,inner sep=2pt]

%style for values taken by X
\tikzstyle{BernGuide} = [inner sep=2pt,text=teal,right=1em]

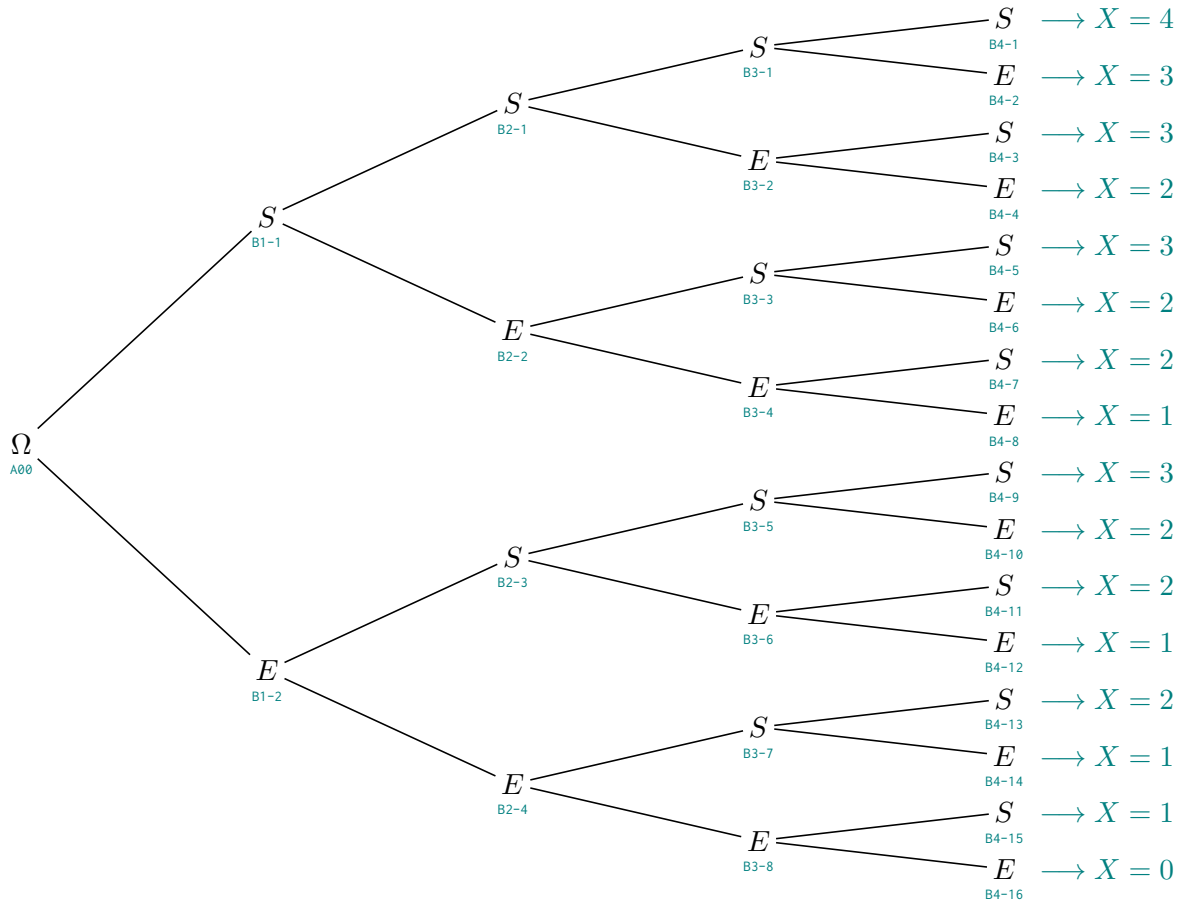
%command to restore default styles
\tkzBernTreeStyleDefault
```

### 3.2 Examples

```
\begin{tikzpicture}
  \tkzBernoulliTree
\end{tikzpicture}
```



`\tkzBernoulliTree*[Help,ShowProbs=false,LevelSep=3.25,NodeSep=0.75,N=4,Guide,var=Z]`



```
%\usepackage{nicefrac}
\tikzstyle{BernEdge} = [thick,red,->]
\tikzstyle{BernRoot} = []
\tikzstyle{BernNodeS} = [draw,rectangle,fill=yellow,inner sep=1.5pt]
\tikzstyle{BernNodeF} = [draw,rectangle,fill=orange,inner sep=1.5pt]
\tikzstyle{BernProbS} = [text=teal,midway,fill=cyan!5,inner sep=1.5pt]
\tikzstyle{BernProbF} = [text=orange,midway,fill=purple!5,inner sep=1.5pt]
\tikzstyle{BernGuide} = [draw,rectangle,inner sep=2pt,text=green,right=2em]
\tikzBernoulliTree*[%
  Events={\A$/\overline{A}$},%
  NodeSep=0.75,Guide,%
  Probs={\nicefrac{1}{6}}/{\nicefrac{5}{6}}]
```

