1 Overview

This package is intended to simplify the drawing of probability trees with \textsc{metaPost}. It provides one main command and several parameters to control the output.

It can be used in standalone files with two compilations (\textsc{ltxmp} package is loaded) but also with \textsc{Lualatex} and \textsc{luamplib} package.

\begin{verbatim}
\texttt{tree[i][j]}(<\texttt{dim1}, <\texttt{dim2}, \ldots><\texttt{ev1}, <\texttt{prob1}, <\texttt{ev2}, <\texttt{prob2}, \ldots>) picture
\end{verbatim}

Probability tree located in column \texttt{i} and row \texttt{j} (see figure below). \texttt{dim1, dim2, \ldots} can be numerics or pairs and control the dimension of the tree. \texttt{ev1, prob1, \ldots} can be strings or pictures and will be printed (using \textsc{ltxmp} if strings) at the end of the edge (the event) and above the edge (the probability).
Note that you can use these commands inside any \texttt{beginfig()};...\texttt{endfig}; but sometimes, for some constructions, they need to be enclosed between \texttt{begintree} and \texttt{endtree} commands. Such commands are indicated with a margin note.

## 2 Trees

### 2.1 Different kinds of trees

\texttt{tree[i][j](width,vspace)(ev1,prob1,ev2,prob2,...) picture}

Regular tree where \texttt{width} is the horizontal width of the tree and \texttt{vspace} the vertical space between two consecutive nodes.

\textbf{Exemple 1}

\begin{verbatim}
beginfig(1);
draw tree[1][1](4 cm,2.5 cm)("$A_1$","$\nicefrac{1}{3}$","$A_2$","$\nicefrac{2}{3}$");
draw tree[2][1](3 cm,1.5 cm)("$B$","$\nicefrac{1}{4}$","$C$","$\nicefrac{3}{4}$");
draw tree[2][2](3 cm,1 cm)("$D$","$p$","$E$","$q$","$F$","$r$");
endfig;
\end{verbatim}

\begin{center}
\begin{tikzpicture}
\node (A1) at (0,0) {$A_1$};
\node (A2) at (0,-2) {$A_2$};
\node (B) at (1.5,1) {$B$};
\node (C) at (1.5,0) {$C$};
\node (D) at (1.5,-1) {$D$};
\node (E) at (1.5,-2) {$E$};
\node (F) at (1.5,-3) {$F$};
\draw (A1) -- (B) node [midway, above, sloped] {\textcolor{red}{\scriptsize{$\nicefrac{1}{3}$}}};
\draw (A1) -- (C) node [midway, above, sloped] {\textcolor{green}{\scriptsize{$\nicefrac{1}{4}$}}};
\draw (A1) -- (D) node [midway, above, sloped] {\textcolor{blue}{\scriptsize{$p$}}};
\draw (A2) -- (E) node [midway, above, sloped] {\textcolor{red}{\scriptsize{$\nicefrac{2}{3}$}}};
\draw (A2) -- (F) node [midway, above, sloped] {\textcolor{green}{\scriptsize{$q$}}};
\draw (A2) -- (F) node [midway, above, sloped] {\textcolor{blue}{\scriptsize{$r$}}};
\draw (A1) -- (B) node [midway, above, sloped] {\textcolor{red}{\scriptsize{$\nicefrac{1}{3}$}}};
\draw (A1) -- (C) node [midway, above, sloped] {\textcolor{green}{\scriptsize{$\nicefrac{1}{4}$}}};
\draw (A1) -- (D) node [midway, above, sloped] {\textcolor{blue}{\scriptsize{$p$}}};
\draw (A2) -- (E) node [midway, above, sloped] {\textcolor{red}{\scriptsize{$\nicefrac{2}{3}$}}};
\draw (A2) -- (F) node [midway, above, sloped] {\textcolor{green}{\scriptsize{$q$}}};
\draw (A2) -- (F) node [midway, above, sloped] {\textcolor{blue}{\scriptsize{$r$}}};
\end{tikzpicture}
\end{center}

\texttt{tree[i][j](width,vsp1,vsp2,...)(ev1,p1,ev2,p2,...) picture}

Tree where \texttt{width} is the horizontal width of the tree while each \texttt{vsp} indicates the vertical space between the node and the origin of the tree.

\textbf{Exemple 2}

\begin{verbatim}
beginfig(2);
draw tree[1][1](3 cm,2 cm)("$A$","$p$","$\overline{A}$","$q$");
draw tree[2][1](3 cm,2 cm,1 cm,-1 cm)("$B$","$p$","$C$","$q$","$D$","$r$");
draw tree[2][2](3 cm,0 cm,-2 cm)("$E$","$0.5$","$F$","$0.5$");
endfig;
\end{verbatim}
Exemple 3

beginfig(3);
  draw tree[1][1](3 cm, 2 cm)("$A$", "$p$", "$\overline{A}$", "$1-p$);
  draw tree[2][1]((3 cm, 2 cm), (4 cm, -1 cm))("$B$", "$q$", "$C$", "$r$);
endfig;

2.2 Simple trees

stree[<i>]<j>](...) picture

Same as previous except that there are no probabilities.

Exemple 4

beginfig(4);
  draw stree[1][1](100, 50)("$A$", "$B$", "$C$);
  draw stree[2][1](80, 25)("$A$", "$B$);
  draw stree[2][2](80, 25)("$A$", "$B$);
  draw stree[2][3](80, 25)("$A$", "$B$);
endfig;
2.3 Start and end labels

\textbf{startlabel}(<s>) \textit{picture}

Prints \textit{s} (can be a string or a picture) at the origin of the tree.

\textbf{Exemple 5}

\begin{verbatim}
beginfig(5);
draw startlabel("S");
draw tree[1][1](3cm,2cm)("A","p","B","q");
endfig;
\end{verbatim}

\textbf{endlabel}[<i>][<j>](<s>) \textit{picture}

Prints \textit{s} at the end of a branch. The space between the previous label and \textit{s} is controlled by the numeric \texttt{endlabelspace} which defaults to 1cm.

\textbf{Exemple 6}

\begin{verbatim}
beginfig(6);
draw startlabel("S");
draw tree[1][1](3cm,2cm)("A","p","B","q");
draw tree[2][2](2cm,1cm)("A","p","B","q");
draw endlabel[2][1]("SA");
draw endlabel[3][1]("SBA");
draw endlabel[3][2]("SBB");
endfig;
\end{verbatim}
3 Direction

dirtree numeric, default: 0

All trees are constructed horizontally by default. dirtree indicates the angle in degrees between the horizontal and the main direction of the tree.

Exemple 7

\begin{verbatim}
beginfig(7);
dirtree:=135;
draw tree[1][1](3cm,2cm)("$A_1$","a_1","A_2","a_2");
draw tree[2][1](3cm,1cm)("B","b","C","c");
draw tree[2][2](3cm,1cm)("D","p","E","q");
endfig;
\end{verbatim}

Exemple 8

\begin{verbatim}
beginfig(8);
dirtree:=-60;
draw tree[1][1](3cm,2cm)("$A_1$","a_1","A_2","a_2");
draw tree[2][1](3cm,1cm)("B","b","C","c");
draw tree[2][2](3cm,1cm)("D","p","E","q");
endfig;
\end{verbatim}
All the trees are viewed as “horizontal” trees, so the space between two subtrees is horizontal too. With \texttt{dirtree}, the whole (horizontal) tree is rotated. But if the tree is designed vertically, spacing is wrong. In this case, one can use \texttt{dirlabel} to indicate the orientation of the tree.

\begin{example}
\begin{figure}
\begin{verbatim}
beginfig(9);
draw tree[1][1]((-1cm,2cm),(1cm,2cm))
  ("$A$","p","$B$","q");
draw tree[2][1]((-0.5cm,2cm),(0.5cm,2cm))
  ("$C$","c","$D$","d");
draw tree[2][2]((-0.5cm,2cm),(0.5cm,2cm))
  ("$E$","e","$F$","f");
endfig;
\end{verbatim}
\end{figure}
\end{example}

\begin{example}
\begin{figure}
\begin{verbatim}
beginfig(10);
dirlabel:=90;
draw tree[1][1]((-1cm,2cm),(1cm,2cm))
  ("$A$","p","$B$","q");
draw tree[2][1]((-0.5cm,2cm),(0.5cm,2cm))
  ("$C$","c","$D$","d");
draw tree[2][2]((-0.5cm,2cm),(0.5cm,2cm))
  ("$E$","e","$F$","f");
endfig;
\end{verbatim}
\end{figure}
\end{example}

4 Dealing with alignment

\begin{example}
\begin{figure}
\begin{verbatim}
beginfig(11);
draw tree[1][1](80,120)("$A$","0.5","\overline{A}$","0.5");
draw tree[2][1](70,40)("Yes","p","No","q","Maybe","r");
draw tree[2][2](70,40,"shiftev:=1.5cm")("Yes","p","No","q","Maybe","r");
draw tree[3][1](50,20)("B","b","C","c");
draw tree[3][2](50,20)("B","b","C","c");
draw tree[3][3](50,20)("B","b","C","c");
draw tree[3][4](50,20)("B","b","C","c");
draw tree[3][5](50,20)("B","b","C","c");
draw tree[3][6](50,20)("B","b","C","c");
endfig;
\end{verbatim}
\end{figure}
\end{example}
With the boolean `abscoord` set to `true`, all the coordinates are given from the origin of the first tree instead of the origin of the subtree, which makes easier the alignment of all the subtrees.

**Exemple 12**

```plaintext
beginfig(12);
abscoord:=true;
draw tree[1][1](3cm,2cm)("A","p","Blabla","q");
draw tree[2][1](7cm,1.5cm),(7cm,0.5cm)("A","p","B","q");
draw tree[2][2](7cm,-0.5cm),(7cm,-1.5cm)("A","p","B","q");
endfig;
```

5 Parameters

All following parameters can be changed globally before drawing the tree or changed locally inside the first set of parameters:

```plaintext
scaleev:=2;
draw tree[1][1](3cm,2cm)(...);
draw tree[2][1](3cm,2cm)(...);
```
or

```plaintext
draw tree[1][1](3cm,2cm,"scaleev:=2")(...);
draw tree[2][1](3cm,2cm)(...);
```

In the first case, `scaleev` is changed globally while in the second case, the change only applies to the first tree.

### 5.1 Event

**scaleev**

Numeric controlling the scale of the label at the end of the edge (the event).

**Exemple 13**

```plaintext
beginfig(13);
scaleev:=2;
draw stree[1][1](3cm,2cm)("$A$","$B$");
endfig;
```

**nodeformat**

String, default: ""

String that indicates how the events are printed (the shape of path around the event). Possible values are (for now) "bbox", "circle", "superellipse".

**Exemple 14**

```plaintext
beginfig(14);
nodeformat:="bbox";
draw stree[1][1](3cm,2cm)("$A$","$B$");
endfig;
```

**nodecolor**

Color, default: black

Color of the path around the node

**nodebgcolor**

Color, default: white

Color of the background of the region delimited by the previous path.

**nodefgcolor**

Color, default: black

Color of the text.
5.2 Leaves

You may want to format the leaves in a different way from the nodes. A tree using the following parameters must be enclosed in a \texttt{begintree;...endtree;} “environment”.

\begin{itemize}
\item \texttt{leaveformat} \texttt{string}, default: \texttt{""}
\item String that indicates how the events are printed (the shape of path around the event). Possible values are (for now) \texttt{"bbox"}, \texttt{"circle"}, \texttt{"superellipse"} and \texttt{"none"}.
\end{itemize}

Exemple 17

\begin{verbatim}
beginfig(17);
begintree;
leaveformat:="bbox";
\draw stree[1][1](100,45)("$A$","$B$");
\draw stree[2][1](80,30)("$C$","$D$");
\draw stree[3][2](65,20)("$E$","$F$");
endtree;
endfig;
\end{verbatim}
**Exemple 18**

```plaintext
beginfig(18);
begintree;
nodeformat:="circle";
nodelinecolor:=(0.8,0,0); nodebgcolor:=(1,0.5,0.5); nodefgcolor:=white;
leaveformat:="bbox";
leavebgcolor:=(0.3,1,1); leavefgcolor:=red;
draw stree[1][1](100,45)("A","B");
draw stree[2][1](80,30)("C","D");
draw stree[3][2](65,20)("E","F");
endtree;
endfig;
```

Note that `nodeformat` applies to both nodes and leaves. To avoid formatting the leaves, use the value "none" for `leaveformat`.

**Exemple 19**

```plaintext
beginfig(19);
begintree;
nodeformat:="circle";
nodelinecolor:=(0.8,0,0); nodebgcolor:=(1,0.5,0.5); nodefgcolor:=white;
leaveformat:="none";
draw stree[1][1](100,45)("A","B");
draw stree[2][1](80,30)("C","D");
draw stree[3][2](65,20)("E","F");
endtree;
endfig;
```

5.3 Probability

**probformat string, default: ""**

String that indicates how the probabilities are printed (the shape of path around the probability). Possible values are (for now) "bbox", "circle", "superellipse".
Exemple 20

beginfig(20);
probformat:="bbox";
draw tree[1][1](3cm,2cm)("A","p","B","q");
endfig;

problinelinecolor color, default: black

Color of the path around the probability

probbgcolor color, default: white

Color of the background of the region delimited by the previous path.

probfgcolor color, default: black

Color of the text.

Exemple 21

beginfig(21);
probformat:="circle";
problinelinecolor:=(0.8,0,0);
probbgcolor:=(1,0.5,0.5);
probfgcolor:=white;
draw tree[1][1](3cm,2cm)("A","p","B","q");
endfig;

Exemple 22

beginfig(22);
probfgcolor:=blue;
draw tree[1][1](3cm,2cm)("A","p","B","q");
endfig;

scaleprob numeric, default: 0.85

Numeric controlling the scale of the label above the edge (the probability).

Exemple 23

beginfig(23);
scaleprob:=1.5;
draw tree[1][1](3cm,2cm)("A","p","B","q");
endfig;

posprob numeric, default: 0.6

Numeric controlling the position of the label above the edge.
**Exemple 24**

```latex
beginfig(24);
pospb:=0.8;
draw tree[1][1](3cm,2cm)("A","p","B","q");
endfig;
```

**typeprob**

*numeric, default: 1*

Numeric controlling how the label is printed. Values can be 1 (the label is printed above the edge), 2 (the label is printed on the edge), 3 (the label is printed above the edge and rotated) or 4 (the label is printed on the edge and rotated).

**Exemple 25**

```latex
beginfig(25);
typetb:=2;
draw tree[1][1](3cm,2cm)("A","p","B","1-p");
endfig;
```

**Exemple 26**

```latex
beginfig(26);
typetb:=3;
draw tree[1][1](3cm,2cm)("A","p","B","1-p");
endfig;
```

**Exemple 27**

```latex
beginfig(27);
typetb:=4;
draw tree[1][1](3cm,2cm)("A","p","B","1-p");
endfig;
```

**proboffset**

*numeric, default: 3bp*

Numeric controlling the amount by which the label above the edge is offset.

**Exemple 28**

```latex
beginfig(28);
draw tree[1][1](3cm,3cm)("A","p+q+r","B","s");
endfig;
```
Exemple 29

```plaintext
beginfig(29);
proboffset:=6bp;
draw tree[1][1](3cm,3cm)("A","p+q+r","B","s");
endfig;
```

5.4 Edge

**linewidth**

numeric, default: 0.5bp

Width of the lines.

**linecolor**

color, default: black

Color of the lines.

Exemple 30

```plaintext
beginfig(30);
linewidth:=1.5;
linecolor:=blue;
draw tree[1][1](3cm,2cm)("A","p","B","q");
endfig;
```

**endedgeshift**

numeric, default: 0

Vertical space added at the end of the edge. Useful when various edges end at the same point.

Exemple 31

```plaintext
beginfig(31);
draw startlabel("S");
draw tree[1][1]((3cm,-1cm))("A","p");
endfig;
```

Exemple 32

```plaintext
beginfig(32);
endedgeshift:=10;
draw startlabel("S");
draw tree[1][1]((3cm,-1cm))("A","p");
endfig;
```

**edgearrow**

boolean, default: false

When the boolean `edgearrow` is set to true, edges end with an arrow.
**Exemple 33**

```plaintext
defedgearrow:=true;
draw tree[1][1](3cm,2cm)("A","p","B","q");
endfig;
```

**branchtype**  
**string, default: "segment"**

String which indicates the shape of the edge. Possible values are segment, curve, broken. Note that double quotes have to be replaced by single quotes when this parameter is changed locally inside the tree macro.

**tenscurve**  
**numeric, default: 0**

If string `branchtype` is set to `curve`, `tenscurve` indicates the “tension”. When set to 1, the curve is a segment.

**Exemple 34**

```plaintext
beginfig(34);
branchtype:="curve";
draw stree[1][1](3cm,3cm)("A","B");
endfig;
```

**Exemple 35**

```plaintext
beginfig(35);
draw stree[1][1](3cm,3cm,"branchtype='curve'")("A","B");
endfig;
```

**Exemple 36**

```plaintext
beginfig(36);
branchtype:="curve";
tenscurve:=0.5;
draw stree[1][1](3cm,3cm)("A","B");
endfig;
```

**brokenlineratio**  
**numeric, default: 0.2**

If string `branchtype` is set to `broken`, `brokenlineratio` indicates the ratio between the length of the first segment of the broken line and the total length of the horizontal space.
6 Regular trees

6.1 Ordinary regular trees

\[ \texttt{regulartree}(n)(l,h)(\texttt{ev1},\texttt{prob1},\texttt{ev2},\texttt{prob2},...) \] picture

Tree describing the repetition of \( n \) identical and independent random experiments. \( l \) is the horizontal length of the first edges and \( h \) is the vertical space between two leaves.

\[ \texttt{scalebranch} \quad \text{numeric, default: 0.8} \]

Ratio between edges width of consecutive level.

Exemple 40

\[ \texttt{beginfig(40); draw regulartree(2)(3cm,0.7cm) \"A\" \"p\" \"B\" \"q\" \"C\" \"r\";) endfig; } \]

Diagram of a regular tree with probabilities and labels.
Note that you can change variable values inside the first set of parameters.

**Exemple 41**

```latex
beginfig(41);
draw regulartree(2)(3cm,0.7cm,"typeprob:=2")
("$A$","$p$","$B$","$q$","$C$","$r$");
endfig;
```

---

### 6.2 Binomial trees

**bernoulliprocess(<n>)(<l>,<h>)(<ev1>,<prob1>,<ev2>,<prob2>) picture**

Tree describing the Bernoulli process with \( n \) trials. \( l \) is the horizontal length of the first edges and \( h \) is the vertical space between two final nodes. If the last set of parameters is omitted, the values are set according to the following parameters.

**bernoulliprocessL(<n>)(<L>,<H>)(<ev1>,<prob1>,<ev2>,<prob2>) picture**

Same as above where \( L \) is the whole width of the tree and \( H \) its height.

Several parameters control the output:

- **bernoullisuccessevent** string, default: "$S$"  
  String printed at every node representing a success.

- **bernoullifailureevent** string, default: "$\overline{S}$"  
  String printed at every node representing a failure.

- **bernoullisuccessprob** string, default: "$p$"  
  String printed above every edge representing a success.

- **bernoullifailureprob** string, default: "$q$"  
  String printed above every edge representing a failure.

- **bernoulliscalebranch** numeric, default: 0.8  
  Ratio between width of consecutive edges.
Exemple 42

```plaintext
beginfig(42);
draw bernoulliprocess(3)(3cm,0.7cm)();
endfig;
```

Exemple 43

```plaintext
beginfig(43);
draw bernoulliprocess(3)(3cm,0.7cm,"typeprob:=2;")
("$A$","$0.7$","$B$","$0.3$");
endfig;
```

Exemple 44

```plaintext
beginfig(44);
typeprob:=4;
bernoullisuccessevent:="$A$";
bernoullifailureevent:="$B$";
bernoullisuccessprob:="0.7$";
bernoullifailureprob:="0.3$";
draw bernoulliprocess(3)(3cm,0.7cm)();
endfig;
```

```plaintext
binomialtree(<n>)(<l>,<h>) picture
Tree describing the binomial distribution with n trials. l is the length of the first edges and h is the space between two final nodes. It uses bernoullisuccesprob and bernoullifailureprob but bernoulliscalebranch is set to 1.
```

```plaintext
binomialtreeL(<n>)(<L>,<H>) picture
Same as above where L is the whole width of the tree and H its height.
```

Exemple 45

```plaintext
beginfig(45);
draw binomialtree(4)(3cm,1.5cm);
endfig;
```
7 “Calculated” trees

The following commands are experimental and need to be enclosed in a \texttt{begintree;...endtree; “environment”}.

\begin{verbatim}
\begin{verbatim}
tree[i][j]()(<ev1>,<prob1>,<ev2>,<prob2>,...)
\end{verbatim}
\end{verbatim}

When the first set of parameters is left empty, the dimensions of the tree are calculated. The calculations use the parameters described below.

\begin{verbatim}
\begin{verbatim}
stree[i][j]()(<ev1>,<ev2>,...)
\end{verbatim}
\end{verbatim}

Same as above for “simple” trees.

\textbf{Exemple 46}

\begin{verbatim}
beginfig(46);
beginintree;
\draw startlabel("S");
\draw stree[1][1]()("A","B","C");
\draw stree[2][1]()("D","E","F");
\draw stree[2][3]()("D","E");
\draw stree[3][2]()("G","H");
\draw stree[3][4]()("G","H");
endtree;
endfig;
\end{verbatim}
widthbranch numeric, default: 3.5cm

Horizontal width of the first level tree.

gapnode numeric, default: 0.7cm

Minimal vertical space between two nodes of the last level trees.

scalebranch numeric, default: 0.8

Ratio between edges width of consecutive level.

Exemple 47

```plaintext
beginfig(47);
beginintree;
widthbranch:=4cm;
scalebranch:=1;
gapnode:=1cm;
draw startlabel("S");
draw stree[1][1]("A","B","C");
draw stree[2][1]("D","E","F");
draw stree[2][3]("D","E");
draw stree[3][2]("G","H");
draw stree[3][4]("G","H");
endtree;
endfig;
```

![Tree Diagram](image-url)
8 Examples

Exemple 48

```latex
beginfig(48);
  u:=0.4cm;
  branchtype:="curve";
  dirlabel:=90;
  abscoord:=true;
  endlabelspace:=0.5cm;
  draw startlabel("S");
  draw stree[1][1]((-5.5u,4u),(5.5u,8u))("NP","VP");
  draw stree[2][1]((-8.5u,12u),(-2.5u,8u))("A","NP");
  draw stree[2][2]((3.5u,12u),(7.5u,12u))("V","Adv");
  draw stree[3][2]((-4.5u,12u),(-0.5u,12u))("A","N");
  draw endlabel[3][1]("Colorless");
  draw endlabel[4][1]("green");
  draw endlabel[4][2]("ideas");
  draw endlabel[3][3]("sleep");
  draw endlabel[3][4]("furiously");
endfig;
```

Exemple 49

```latex
beginfig(49);
  u:=1cm;
  branchtype:="broken";
  dirlabel:=-90;
  abscoord:=true;
  scaleev:=2;
  label.top(textext("\Large Tree diagram of $(2x+1)(x-3)$"),(0,1cm));
  draw startlabel("$\times$");
  draw stree[1][1]((-2u,-1.5u),(2u,-1.5u))("+$","-$");
  draw stree[2][1]((-3u,-3.5u),(-1u,-3.5u))("$\times$","$1$");
  draw stree[2][2]((1u,-3.5u),(3u,-3.5u))("$x$","$3$");
  draw stree[3][1]((-4u,-5.5u),(-2u,-5.5u))("$\times$","$2$");
endfig;
```
Tree diagram of $(2x + 1)(x - 3)$

Exemple 50

```latex
beginfig(50);
posprob:=0.5;
typeprob:=3;
shiftev:=1.5cm;
edgearrow:=true;
u:=0.2cm;
vardef paral = ((2,-2)--(6,2)--(0,2)--(-4,-2)--cycle) scaled u enddef;
vardef rhombus = ((3,0)--(0,6)--(-3,0)--(0,-6)--cycle) scaled u enddef;
vardef rectangle = ((3,5)--(-3,5)--(-3,-5)--(3,-5)--cycle) scaled u enddef;
vardef square = ((3,3)--(-3,3)--(-3,-3)--(3,-3)--cycle) scaled u enddef;
draw startlabel(paral);
draw tree[1][1](5cm,4cm)(rhombus,"Diagonals perpendicular",% rectangle,"Diagonals of equal length");
endedgeshift:=5;
draw tree[2][1]((5cm,-2cm))("","Diagonals of equal length");
draw tree[2][2]((5cm,2cm))(square,"Diagonals perpendicular");
endfig;
```
beginfig(51);
dirtree:=-90;
branchtype:="curve"; tenscurve:=0.75;
linewidth:=1; linecolor:=(0.2,0.2,0.7);
widthbranch:=1cm; scalebranch:=0.9;
gapnode:=1cm;
leaveformat:="bbox";
nodeformat:="superellipse"; nodebgcolor:=(0.6,0.6,1);
begintree;
label.top(textext("\Large Huffman tree (source Wikipedia)"),(0,1cm));
draw startlabel("36");
draw stree[1][1]("20","16");
draw stree[2][1]("12","8");
draw stree[2][2]("8","8");
draw stree[3][1]("7","5");
draw stree[3][2]("4","4");
draw stree[3][3]("4","a\,4");
draw stree[3][4]("4","e","4");
draw stree[4][2]("f","3","2");
draw stree[4][3]("s","h","2");
draw stree[4][4]("2","i","2");
draw stree[4][5]("m","t","2");
draw stree[5][2]("l","r","1");
draw stree[5][5]("p","x","1");
draw stree[5][9]("u","o","1");
endtree;
endfig;

Huffman tree (source Wikipedia)