```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% GasTeX : Graphs and Automata Simplified in TeX
%%
%% Macros for drawing easily graphs and automata under the picture
%% environment of LaTeX2e.
%% Documentation in gastex.doc
%% See also the comments before the macros throughout the file.
%% Examples in ex_gastex.tex
%%
%% Paul Gastin
%% LIAFA
%% Universite Paris 7
%% 2, place Jussieu
%% F-75251 Paris Cedex 05
%% email : Paul.Gastin@liafa.jussieu.fr
%% www : http://www.liafa.jussieu.fr/~gastin
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% A lot of changes has been made since gastex 1.0 and as a result,
% these new macros are no longer compatible with the previous ones.
% To be able to use old pictures, a compatible mode is provided
% (see \compatiblegastexun at the end of the file). The compatibility
% is almost 100% and should be sufficient in most cases.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% History since version 2.0
%
% 2.01:
% - Fix an error that occured in v2.0 when using Nfill=y without
% defining previously fillgray or fillcolor.
% The following default setting has been added.
% \gasset{fillgray=0,Nfill=n} % Not filled but black if filled
%
% 2.1:
% - New macros to draw directly circles, rectanges, ovals,
% lines and bezier curves.
%   \drawcircle, \drawrect, \drawoval,
%   \drawline, \drawqbezier, \drawcbezier
% All these macros uses gasset options and in particular:
% Nframe, Nfill, linecolor, fillcolor, dash, AHnb, etc...
% - Compatibility mode for pspictpg up to v0.6
% - Fix the bug which occured sometimes when using Nw=0,Nh=0.
%
% 2.2:
% - added the options slide and paper to the package.
% In order to get the default settings for slides, use
%   \usepackage[slide]{gastex}
% The default settings for papers is obtained with
%   \usepackage[paper]{gastex}
% or
%   \usepackage{gastex}
% - added new option loopCW to define whether loops are in
% clockwise direction or not.
% - Fix a TeX error (Arithmetic overflow) that occurred when using
```

```
% \drawedge(A,B){} with two nodes A and B having the same coordinates.
% Now, in this case, an error message is issued in the log
% and the macro \drawedge(A,B){} is ignored.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Known problems and (hopefully) solutions.
%
% 21/10/99: Frank Goertzen (frank.goertzen@unibw-muenchen.de) has
% reported that when using gastex together with german an error may
% occur when running dvips if the german package is loaded first.  The
% solution is to load gastex before german:
% \usepackage{gastex}
% \usepackage{german}
% I have no idea concerning the cause of this error.
%
% 07/03/00: A postscript error may occur when using Nw=0,Nh=0.
% Solution: Use a small value instead of 0, e.g. Nw=0.1,Nh=0.1
% Fixed in version 2.1.
%
% 27/10/00: Using gasset inside a tabular or an array produces an error.
% The reason is that I'm using the "&" symbol as a marker in order to
% process gasset options.
% Solution: Include the whole picture inside an mbox.
% \begin{tabular}{c}
%   \mbox{\begin{picture}(10,20)(-5,-5)
%   \gasset{ELdist=0}
%   \node(A)(0,0){1}\drawloop(A){$a$}
%   \end{picture}}
% \end{tabular}
%
% pdflatex: gastex does not work with pdflatex since it produces
% postscript code and pdflatex does not know what to do with it.
% Solution: Use latex and then ps2pdf.
% I don't know whether it is possible to translate my postscript code
% into pdf code. The problem is that I'm using postscript to make some
% computations and not only to draw the picture.
% I would appreciate the help of a pdf guru on this.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Settings
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%-----------------------------------------------------------------------
% Set values for gastex parameters.
%
%   \gasset{parameter=value,parameter=value,...}
%   optional spaces are only allowed after commas
%
% Nw=number
%   widht in \unitlength of oval nodes
% Nh=number
%   height in \unitlength of oval nodes
% Nmr=number
%   maximal radius in \unitlength of oval nodes
```

```
%   The actual radius will be min(Nw/2,Nh/2,Nmr)
%   Nmr=0 yields a rectangle node
%
% Nadjust=any combination of the letters w,h,n
%   w : the node width is adjusted to the node label width + Nadjustdist
%   h : the node height is adjusted to the node label height +
Nadjustdist
%   n : no adjustment
% Nadjustdist=number
%   distance in \unitlength between the label box and the node frame
%   when using Nadjust
%
% Nframe=y or n
%   boolean flag defining whether nodes are framed or not.
% Nfill=y or n
%   boolean flag defining whether nodes are filled or not.
%   Automatically set to true by the parameters fillgray and fillcolor
%
% ExtNL=y or n
%   boolean flag defining whether node labels are external or not.
% NLangle=number
%   direction in degree of node labels
% NLdist=number
%   The meaning depends on the ExtNL flag.
%   If the flag is set to "n" then NLdist defines the distance
%   in \unitlength between the node center and the label center.
%   If the flag is set to "y" then NLdist defines the distance
%   in \unitlength between the node frame and the label box.
%
% Nmarks=any combination of the letters i,f,r,n
%   i : initial mark on nodes
%   f : final mark on nodes
%   r : repeated mark on nodes
%   n : no mark on nodes
% ilength=number
%   length in \unitlength of arrows marking initial nodes
% iangle=number
%   direction in degree of arrows marking initial nodes
% flength=number
%   length in \unitlength of arrows marking final nodes
% fangle=number
%   direction in degree of arrows marking final nodes
% rdist=number
%   distance in \unitlength between the lines of repeated nodes
%   a positive value draw the second line inside the normal one
%   a negative value draw the second line outside the normal one
%   but the edges are still connected to the normal line.
%
% loopdiam=number
%   diameters in \unitlength of loops
% loopangle=number
%   direction in degree of loops
% loopCW=y or n
%   boolean flag defining whether loops are in clockwise direction or
not.
%
% curvedepth=number
```

```
%   depth in \unitlength of curved edges between two nodes
%   The absolute value of curvedepth defines the distance between the
%   middle of the curved edge and the center of the line joining the two
nodes
%   With a positive/negative value the curved edge is on the left/right
%   of the line joining the two nodes.
%
% AHnb=number
%   number of arrowhead(s) at the end of edges.
%   0 for no arrowhead.
% AHdist=number
%   distance in \unitlength between two arrowheads
% AHangle=angle
%   angle in degree between the edge and the arrowhead side
% AHLength=number
%   Length in \unitlegth of the arrowhead side
% AHlength=number
%   length in \unitlegth defining the shape of the arrowhead
%     0 for an arrowhead formed with just two lines
%     Length*cos(angle) for a triangular arrowhead
%     See examples.
%
% ELside=l or r
%   label on the (l)eft or (r)ight side of the edge
% ELpos=0..100
%   position of the label along the edge.
%     0 : starting node
%    50 : middle of the edge
%   100 : ending node
% ELdist=number
%   distance in \unitlength between the label and the edge
%
% linegray=decimal number between 0 and 1
%   gray level used to draw lines. 0=black, 1=white.
% fillgray=decimal number between 0 and 1
%   gray level used to fill nodes. 0=black, 1=white.
% linecolor=ColorName
%   color used to draw lines.
%   The color name should be defined in dvipsnam.def and one should
%   include \usepackage[usenames]{color}.
%   This is to avoid the trouble of defining our own colors.
%   The drawback is that it is not possible to define and use other
colors.
%   It should not be very restrictive since plenty of colours are
%   defined in dvipsnam.def.
%   It should not be difficult to add the possibility of using new
%   colors if needed.
% fillcolor=ColorName
%   color used to fill nodes. See remarks above.
% linewidth=number
%   width in \unitlegth of lines
% dash={list of numbers}{offset}
%   Set the dash pattern used for drawing postscript paths.
%   The numbers in the list indicate alternatively lengths
%   in \unitlength of dashes and lengths in \unitlength of spaces.
%   The list of lenghts is used circularly.
%   offset allows to start the pattern at some distance from its
```

```
beginning.
%   Here are some examples:
%      dash={}{0} % continuous path
%      dash={1.5}0 % dashs of length 1.5 and empty spaces of length 1.5
%      dash={0.2 0.5}0 % looks like a sequence of dots
%      dash={4 1 1 1}0 % alternation of long and short dashs
%      dash={1.5}{1.5} % we start with the empty space and not the dash
%      dash={4}{2} % we start in the middle of the first dash

%-----------------------------------------------------------------------
% Default settings
\unitlength=1mm
\gasset{Nw=8,Nh=8,Nmr=4} % circle
\gasset{Nframe=y}
\gasset{fillgray=0,Nfill=n} % Not filled but black if filled
\gasset{ExtNL=n,NLangle=90,NLdist=0}
\gasset{iangle=180,ilength=5}
\gasset{fangle=0,flength=5}
\gasset{rdist=0.7}
\gasset{Nmarks=n} % no mark
\gasset{Nadjustdist=1,Nadjust=n} % no adjust
\gasset{loopdiam=8,loopangle=90}
\gasset{curvedepth=0}
% One triangular small arrowhead
\gasset{AHnb=1,AHdist=1.41,AHangle=20,AHLength=1.5,AHlength=1.41}
\gasset{ELside=l,ELpos=50,ELdist=1}
\gasset{linegray=0} % black lines
\gasset{linewidth=0.14,dash={}0} % continuous path

% Settings for slides
\ifgastexslide
  \gasset{Nw=12,Nh=12,Nmr=6,ilength=8,flength=8,rdist=1,loopdiam=12}
  \gasset{linewidth=0.21,AHdist=2.1,AHLength=2.25,AHlength=2.1}
\fi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Nodes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%-----------------------------------------------------------------------
% Define and draw a node.
%
%   \node(NodeName)(x,y){NodeLabel}
%   \node[parameter=value,...](NodeName)(x,y){NodeLabel}
%
% Required arguments:
%   NodeName : name of the node
%   (x,y) : coordinates of the node in \unitlengh.
%   NodeLabel : label of the node. Empty if no label.
% Optional argument:
%   [parameter=value,...]
% One may just define a node without drawing it using an empty label
% and the parameters Nframe=n,Nfill=n

%-----------------------------------------------------------------------
% Add an ingoing arrow to mark an initial node.
% This arrow is usually drawn by \node using Nmarks=i.
```

```
% This separate macro can be used to draw several arrows
% or arrows with different colors, thickness, dash, ...
%
%   \imark(NodeName)
%   \imark[parameter=value,...](NodeName)
%
% Required arguments:
%   NodeName : name of the node
% Optional argument:
%   [parameter=value,...]

%------------------------------------------------------------------------
% Add an outgoing arrow to mark a final node.
% This arrow is usually drawn by \node using Nmarks=f.
% This separate macro can be used to draw several arrows
% or arrows with different colors, thickness, dash, ...
%
%   \fmark(NodeName)
%   \fmark[parameter=value,...](NodeName)
%
% Required arguments:
%   NodeName : name of the node
% Optional argument:
%   [parameter=value,...]

%------------------------------------------------------------------------
% Add a second line to mark a repeated node.
% This line is usually drawn by \node using Nmarks=r.
% This separate macro can be used to draw several lines around the
% node or lines with different colors, thickness, dash, ...
%
%   \rmark(NodeName)
%   \rmark[parameter=value,...](NodeName)
%
% Required arguments:
%   NodeName : name of the node
% Optional argument:
%   [parameter=value,...]

%------------------------------------------------------------------------
% Add a label to the node.
% The position of the label is determined by the ExtNL flag
% and the parameters NLangle and NLdist.
% The label is usually drawn by \node.
% This separate macro can be used to add several labels
%
%   \nodelabel(NodeName){NodeLabel}
%   \nodelabel[parameter=value,...](NodeName){NodeLabel}
%
% Required arguments:
%   NodeName : name of the node
%   NodeLabel : label of the node.
% Optional argument:
%   [parameter=value,...]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Edges
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%------------------------------------------------------------------------
% Edge between two nodes.
%
%   \drawedge(startingNode,endingNode){label}
%   \drawedge[parameter=value,...](startingNode,endingNode){label}
%
% Required arguments:
%   startingNode : name of the starting node,
%   endingNode : name of the ending node,
%   label : label of the edge. {} for no label.
% Optional argument:
%   [parameter=value,...]

%------------------------------------------------------------------------
% Edge between two nodes following a quadratic Bezier curve.
% The first and last control points are the centers of the starting
% and ending nodes resp.
% The middle control point is given by its absolute cartesian coordinates
%
%   \drawqbedge(startingNode,x,y,endingNode){label}
%   \drawqbedge[parameter=value,...](startingNode,x,y,endingNode){label}
%
% Required arguments:
%   startingNode : name of the starting node,
%   x,y : coordinates in \unitlength of the intermediary control point
%   endingNode : name of the ending node,
%   label : label of the edge. {} for no label.
% Optional argument:
%   [parameter=value,...]

%------------------------------------------------------------------------
% Edge between two nodes following a cubic Bezier curve.
% The first and last control points are the centers of the starting
% and ending nodes resp.
% The second and third control points are given by their polar
coordinates
% relative to the centers of the starting and ending nodes resp.
%
%   \drawbpedge(startingNode,sa,sr,endingNode,ea,er){label}
%   \drawbpedge[parameter=value,...](startingNode,sa,sr,endingNode,ea,er)
{label}
%
% Required arguments:
%   startingNode : name of the starting node,
%   sa,sr : polar coordinates of the second control point relative to the
center
%           of the starting node (angle in degree and radius in
\unitlength)
%   endingNode : name of the ending node,
%   ea,er : polar coordinates of the third control point relative to the
center
%           of the ending node (angle in degree and radius in
\unitlength)
%   label : label of the edge. {} for no label.
% Optional argument:
```

```
%   [parameter=value,...]

%-----------------------------------------------------------------------
% Edge between two nodes following a cubic Bezier curve.
% The first and last control points are the centers of the starting
% and ending nodes resp.
% The second and third control points are given by their absolute
% cartesian coordinates.
%
%   \drawbcedge(startingNode,xs,ys,endingNode,xe,ye){label}
%   \drawbcedge[parameter=value,...](startingNode,xs,ys,endingNode,xe,ye)
{label}
%
% Required arguments:
%   startingNode : name of the starting node,
%   xs,ys : coordinates in \unitlength of the control point defining
%           the tangent at the starting node
%   endingNode : name of the ending node,
%   xe,ye : coordinates in \unitlength of the control point defining
%           the tangent at the ending node
%   label : label of the edge. {} for no label.
% Optional argument:
%   [parameter=value,...]

%-----------------------------------------------------------------------
% Loop on a node following a cubic Bezier curve.
%
%   \drawloop(Node){label}
%   \drawloop[parameter=value,...](Node){label}
%
% Required arguments:
%   Node : name of the node,
%   label : label of the edge. {} for no label.
% Optional argument:
%   [parameter=value,...]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Other macros
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%-----------------------------------------------------------------------
% Draw a circle.
%
%   \drawcircle(x,y,r)
%   \drawcircle[parameter=value,...](x,y,d)
%
% Required arguments in \unitlengh:
%   x,y : coordinates of the circle center.
%   d : diameter of the circle.
% Optional argument:
%   [parameter=value,...]

%-----------------------------------------------------------------------
% Draw a rectangle.
%
%   \drawrect(x0,y0,x1,y1)
%   \drawrect[parameter=value,...](x0,y0,x1,y1)
```

```
%
% Required arguments in \unitlengh:
%   x0,y0 : coordinates of the lower left corner of the rectangle.
%   x1,y1 : coordinates of the upper right corner of the rectangle.
% Optional argument:
%   [parameter=value,...]

%----------------------------------------------------------------------
% Draw an oval.
%
%   \drawoval(x,y,w,h,mr)
%   \drawoval[parameter=value,...](x,y,w,h,mr)
%
% Required arguments in \unitlengh:
%   x,y : coordinates of the oval center.
%   w,h : width and height of the oval.
%   rm : defines the maximal radius for the corners.
% Optional argument:
%   [parameter=value,...]

%----------------------------------------------------------------------
% Draw a straight line.
% The line may have arrowhead(s), ... depending of the current settings
%
%   \drawline(x0,y0,x1,y1)
%   \drawline[parameter=value,...](x0,y0,x1,y1)
%
% Required arguments:
%   x0,y0,x1,y1 : coordinates in \unitlength of the extremities of the
line
% Optional argument:
%   [parameter=value,...]

%----------------------------------------------------------------------
% Draw a quadratic Bezier curve.
% The line may have arrowhead(s), ... depending of the current settings
%
%   \drawqbezier(x0,y0,x1,y1,x2,y2)
%   \gasline[parameter=value,...](x0,y0,x1,y1,x2,y2)
%
% \drawqbezier arguments:
%   x0,y0,x1,y1,x2,y2 : coordinates in \unitlength of the control points
% Optional argument:
%   [parameter=value,...]

%----------------------------------------------------------------------
% Draw a cubic Bezier curve.
% The line may have arrowhead(s), ... depending of the current settings
%
%   \drawcbezier(x0,y0,x1,y1,x2,y2,x3,y3)
%   \gasline[parameter=value,...](x0,y0,x1,y1,x2,y2,x3,y3)
%
% \drawcbezier arguments:
%   x0,y0,x1,y1,x2,y2,x3,y3 : coordinates in \unitlength of the control
points
% Optional argument:
%   [parameter=value,...]
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Compatibility with gastex v1.0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Should be called inside the scope of a group
%
%   {\compatiblegastexun ... }
%
% The macros \drawloop and \drawedge are redefined to match the old
% ones (v1.0) hence the new ones (v2.0) cannot be used inside the scope
% of \compatiblegastexun.
%
% The compatibility is almost 100%:
% - the following are ignored
%   \setmaxbezier
%   The optional argument of letstate defining the repeated state
diameter
% - \settransdecal, \setedgedecal give curved edges
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Compatibility with pspictpg v0.6
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Should be called inside the scope of a group
%
%   {\compatiblepspictpg ... }
%
% The macros \drawline and \drawcircle are redefined to match those of
% pspictpg hence thoses of GasTeX cannot be used inside the scope
% of \compatiblepspictpg.
%
% The following macros are ignored. Use linecolor and fillcolor instead.
% \pcolor, \pictcolor
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```