NAME

circo − preprocessor for circular layout of graphs

SYNOPSIS

circo [**−G***name*=*value*] [**−N***name*=*value*] [−E*name*=*value*] [**−T***lang*] [**−l** *libfile*] [**−o** *outfile*] [**−vV**] [files]

DESCRIPTION

circo draws graphs using a circular layout (see Six and Tollis, GD ’99 and ALENEX ’99, and Kaufmann and Wiese, GD ’02.) The tool identifies biconnected components and draws the nodes of the component on a circle. The block-cutpoint tree is then laid out using a recursive radial algorithm. Edge crossings within a circle are minimized by placing as many edges on the circle’s perimeter as possible. In particular, if the component is outerplanar, the component will have a planar layout.

If a node belongs to multiple non-trivial biconnected components, the layout puts the node in one of them. By default, this is the first non-trivial component found in the search from the root component.

Input files must be formatted in the *dot* attributed graph language. By default, the output of *circo* is the input graph with layout coordinates appended. To make PostScript, use the **−Tps** option. FrameMaker MIF (**-Tmif**), HPGL (**-Thpgl**), JPEG (**-Tjpeg**), SVG (**-Tsvg**), and GIF (**-Tgif**) are other choices.

Here is a brief synopsis of the graph language.

graph *name* **{** *statement-list* **}** is the top level graph. Statements may be:

name=*val***;**
node [*name*=*val***];**
edge [*name*=*val***];** Set the default graph, node, or edge attribute *name* to *val*. Any subgraph, node, or edge specified after one of these statements inherits these attributes.

n0 [*name0=val0,name1=val1,...***];** Creates node **n0** if it does not exist, and sets its attributes according to the optional list.

n0 −− n1 −− ... **−− nn [***name0=val0,name1=val1,...***];** Creates edges between nodes **n0**, **n1**, ..., **nn** and optionally sets the given attributes. Creates nodes as necessary.

subgraph *name* **{** *statement-list* **}** Creates a subgraph. A subgraph may appear in place of an individual node within an edge statement. The **subgraph** *name* part is optional. If missing, the subgraph is given an internal name.

While attribute names and values may be arbitrary strings, certain fixed attributes control *circo*’s layout algorithm, as next described.

GRAPH ATTRIBUTES

Refer to *dot*(1) for options to control the layout size. In addition, *circo* recognizes the following:

root=*nodename*. Specifies the name of a node occurring in the root block. If the graph is disconnected, the **root** node attribute can be used to specify additional root blocks.

mindist=*value*. Sets the minimum separation between all nodes. If not specified then *circo* uses a default value of 1.0.

splines=*true/false*. If set to true, *circo* will use the graphviz path planning library to draw edges as splines avoiding nodes. If the value is false, or some nodes overlap, edges are drawn as straight line segments connecting nodes. This is also the default style.

NODE ATTRIBUTES

root=*true/false*. This specifies that the block containing the given node be treated as the root of the spanning tree in the layout.

Refer to *dot*(1) for options to control node labels, shapes, sizes, colors, fonts, etc.

EDGE ATTRIBUTES

Refer to *dot*(1) for options to control edge line style and labels.

COMMAND LINE OPTIONS

–V (version) prints version information and exits.

BUGS

AUTHOR

Emden R. Gansner <erg@research.att.com>

SEE ALSO

dot(1), **fdp**(1) **neato**(1) **twopi**(1)