NAME

Hypercube − Graph visualization tool

SYNOPSIS

hypercube−cli [*OPTIONS*] *FILE*
hypercube [*FILE...*]

DESCRIPTION

Hypercube draws text based graph representations as vector images. A simulated annealing based algorithm allowing layout parametrization is used to lay out the graph. **h ypercube-cli** is a command line utility version of Hypercube, **hypercube** a GUI application.

Graphs can be represented as DOT files, GML files, GraphML files, GXL files, edge lists or adjancency matrixes and can be visualized to SVG or EPS images. For info on the file formats, see the **INPUT FORMATS** section.

OPTIONS

Generic Program Information

−h Print a short usage info and exit.

−v Print the program version and exit.

Input and Output Control

−o *FILE*
Set output file to *FILE*. If no output file name is set, it is generated from the input file by replacing its suffix with the output format suffix.

−f *FORMAT*
Set output format to *FORMAT*. Supported formats are**svg** and **eps**. Def ault format is svg.

−e *ENCODING*
Set input file encoding to *ENCODING*. Supported encodings are**iso−8859−1**, **iso−8859−2**, **iso−8859−5**, **iso−8859−7**, **windows−1250**, **windows−1251**, **windows−1252**, **windows−1253**, **koi8−r**, **koi8−u** and **utf−8**. Default encoding is iso−8859−1.

−va *ATTRIBUTE*
Use vertex *ATTRIBUTE* for vertex labels. Usable only with file formats, that support vertex attributes. If no attribute is set, the first available attribute is used.

−ea *ATTRIBUTE*
Use edge *ATTRIBUTE* for edge labels. Usable only with file formats, that support edge attributes. If no attribute is set, the first available attribute is used.

Graph Appearance

−s *SIZE*
Set image size to *SIZE*. The image size is expected as **width,height**.

−d Directed graph − the graph edges are drawn as arrows displaying the edge orientation. If set, overrides the type given in the graph source file.

−u Undirected graph. If set, overrides the type given in the graph source file.

−vc *COLOR*
Set vertex color to*COLOR*. The color format is**#RRGGBB**.

−ec *COLOR*
Set edge color to*COLOR*. The color format is**#RRGGBB**.

−vs *SIZE*
Set vertex size to *SIZE*.

−es *SIZE*
Set edge size to *SIZE*.

−vf *SIZE*
Set vertex ID font size to *SIZE*. To disable showing vertex IDs, set their font size to 0.

−ef *SIZE*
Set edge value font size to *SIZE*. To disable showing edge values, set their font size to 0.

−c Colorize graph. Asign a unique color to every uniqe edge value. When this option is set, the **−ec** option is ommited.

−l *SIZE*
Show edge color legend with font size *SIZE*. Implies the**−c** option.

All sizes are given in output format units − pixels for SVG and points for EPS.

Graph Layout

−nd *DIST*
Set node distribution factor to *DIST*.

−el *LENGTH*
Set edge length factor to *LENGTH*.

−cr *CROSSINGS*
Set edge crossings factor to*CR OSSINGS*.

Algorithm Settings

−it *TEMP*
Set initial temerature to*TEMP*.

−ft *TEMP*
Set final temerature to*TEMP*.

−cf *FACTOR*
Set cooling factor to *FACTOR*.

−ns *STEPS*
Set number of iteration steps to *STEPS*.

INPUT FORMATS

DOT

DOT file format as described in the official Graphviz documentation. Hypercube parses the complete language, but the only used attribute is the label attribute (for both edges and nodes).

```
Digraph {
  0 -> 1 [label = 1];
  0 -> 3 [label = 2];
  0 -> 5 [label = 3];
  1 -> 2 [label = 4];
  1 -> 6 [label = 5];
  2 -> 3 [label = 6];
  2 -> 7 [label = 7];
  3 -> 4 [label = 8];
  4 -> 5 [label = 9];
  4 -> 7 [label = 10];
  5 -> 6 [label = 11];
  6 -> 7 [label = 12];
}
```

GML

GML file format as described in the official documentation. The attributes (keys) used for vertex/edge labels can be set using the **−va** and **−ea** parameters.

```
graph [
  directed 1
```

```
  node [id 0]
  node [id 1]
  node [id 2]
  node [id 3]
  node [id 4]
  node [id 5]
  node [id 6]
  node [id 7]

  edge [label "1" source 0 target 1]
  edge [label "2" source 0 target 3]
  edge [label "3" source 0 target 5]
  edge [label "4" source 1 target 2]
  edge [label "5" source 1 target 6]
  edge [label "6" source 2 target 3]
  edge [label "7" source 2 target 7]
  edge [label "8" source 3 target 4]
  edge [label "9" source 4 target 5]
  edge [label "10" source 4 target 7]
  edge [label "11" source 5 target 6]
  edge [label "12" source 6 target 7]
]
```

GraphML

GraphML file format as given by the specification. Hypercube does not support nested graphs, hyperedges and ports. The attributes (data elements) used for the vertex/edge labels can be set using the **−va** and **−ea** parameters. If there is no appropriate attribute, the vertex/edge id is used as the label.

The input encoding is always taken from the xml declaration (with UTF−8 as the default), setting the encoding using the **−e** parameter is pointless for GraphML files.

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
    http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <graph id="G" edgedefault="directed">
    <node id="0"/>
    <node id="1"/>
    <node id="2"/>
    <node id="3"/>
    <node id="4"/>
    <node id="5"/>
    <node id="6"/>
    <node id="7"/>
    <edge source="0" target="1" id="1"/>
    <edge source="0" target="3" id="2"/>
    <edge source="0" target="5" id="3"/>
    <edge source="1" target="2" id="4"/>
    <edge source="1" target="6" id="5"/>
    <edge source="2" target="3" id="6"/>
    <edge source="2" target="7" id="7"/>
    <edge source="3" target="4" id="8"/>
    <edge source="4" target="5" id="9"/>
    <edge source="4" target="7" id="10"/>
    <edge source="5" target="6" id="11"/>
```

```
    <edge source="6" target="7" id="12"/>
  </graph>
</graphml>
```

GXL

GXL file format as given by the specification. Hypercube does not support hypergraphs, hyperedges and mixed graphs. The attributes used for the vertex/edge labels can be set using the **−va** and **−ea** parameters. If there is no appropriate attribute, the vertex/edge id is used as the label. Composite attribute types (seq, set, bag and tup) are serialized by hypercube into a single string (comma delimited), the locator attribute type is not supported.

The input encoding is always taken from the xml declaration (with UTF−8 as the default), setting the encoding using the **−e** parameter is pointless for GXL files.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE gxl SYSTEM "http://www.gupro.de/GXL/gxl-1.0.dtd">
<gxl>
  <graph>
    <node id="0"/>
    <node id="1"/>
    <node id="2"/>
    <node id="3"/>
    <node id="4"/>
    <node id="5"/>
    <node id="6"/>
    <node id="7"/>
    <edge from="0" to="1" id="1"/>
    <edge from="0" to="3" id="2"/>
    <edge from="0" to="5" id="3"/>
    <edge from="1" to="2" id="4"/>
    <edge from="1" to="6" id="5"/>
    <edge from="2" to="3" id="6"/>
    <edge from="2" to="7" id="7"/>
    <edge from="3" to="4" id="8"/>
    <edge from="4" to="5" id="9"/>
    <edge from="4" to="7" id="10"/>
    <edge from="5" to="6" id="11"/>
    <edge from="6" to="7" id="12"/>
  </graph>
</gxl>
```

Edge list

Each line of the input file represents an edge entry. The first value is the edge's start vertex ID, the second value the end vertex ID and the optional third value is the edge label (value). The values are strings separated by an arbitrary amount of whitespace. Quoted strings can be used, if whitespace occurs in the value.

```
0 1 1
0 3 2
0 5 3
1 2 4
1 6 5
2 3 6
2 7 7
3 4 8
4 5 9
4 7 10
5 6 11
```

6 7 12

Adjacency matrix

The adjacency matrix representation starts with a single number on a separate line representing the number of vertexes of the graph. Starting with the next line, the adjacency matrix itself follows. Numbers greater than 0 are taken as edges with the given edge value.

8
0 1 0 2 0 3 0 0
0 0 4 0 0 0 5 0
0 0 0 6 0 0 0 7
0 0 0 0 8 0 0 0
0 0 0 0 0 9 0 10
0 0 0 0 0 0 11 0
0 0 0 0 0 0 0 12
0 0 0 0 0 0 0 0

SEE ALSO

dot(1), eps2pdf(1), convert(1)

AUTHOR

Martin Tuma (tumic@cbox.cz)