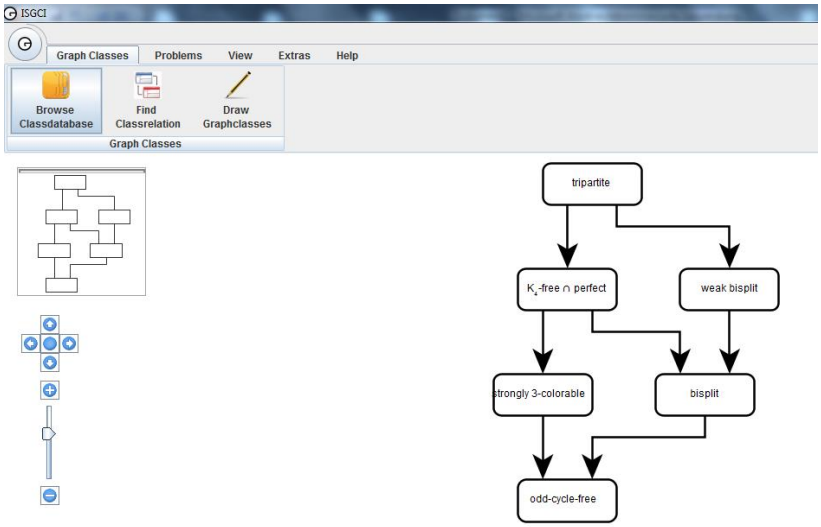
User Interface Documentation

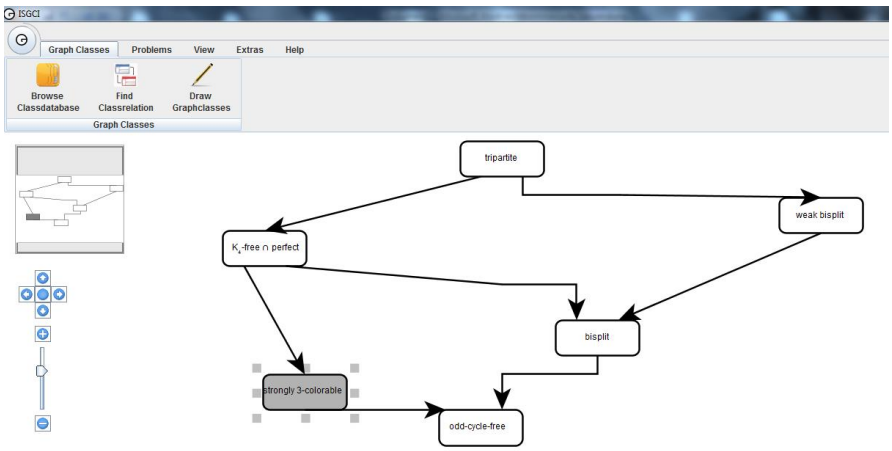
Main window:

The main window is similar in structure to the existing software. However, the menu structure has been replaced with a ribbon, and the canvas (now based on the yFiles library) offers more functionality than before.

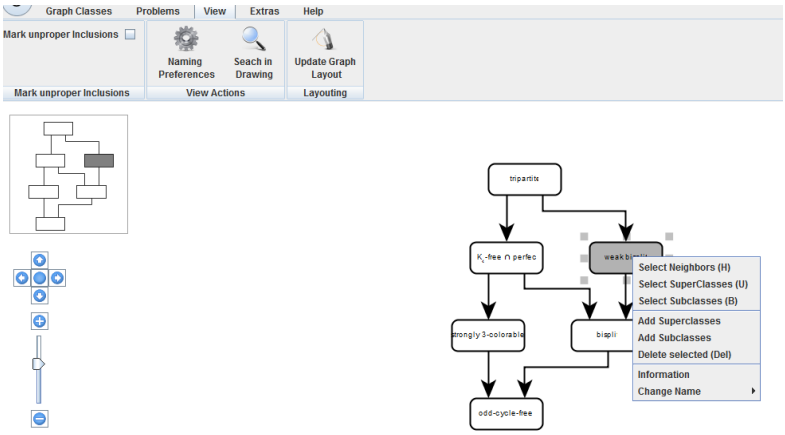
As in the previous version, superclasses are drawn above their subclasses, with a directed arrow going from superclass to subclass



Drag a graph class or an edge to move it.



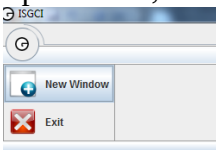
Right-click on a graph class or on an edge will display a menu for the class or inclusion.

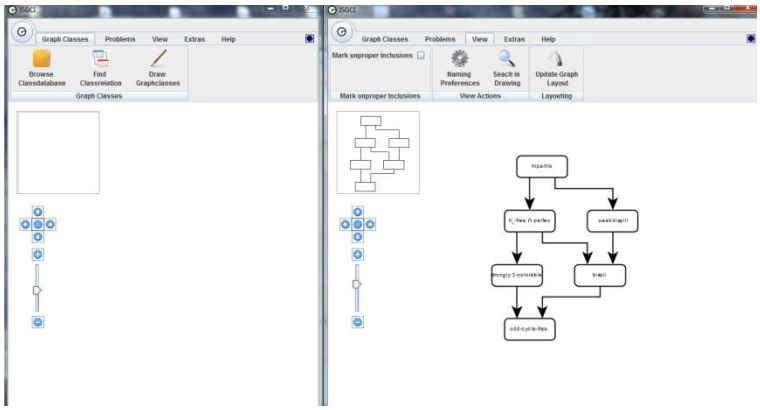


File menu:

<u>New window:</u>

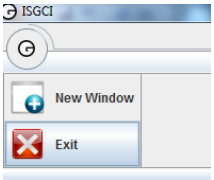
Open a new, independent ISGCI window.





<u>Exit:</u>

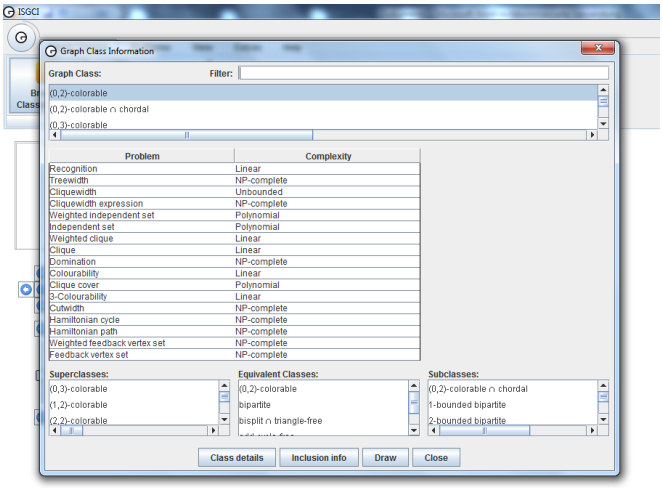
Close the current windows and exit ISGCI.



Graph Classes menu:

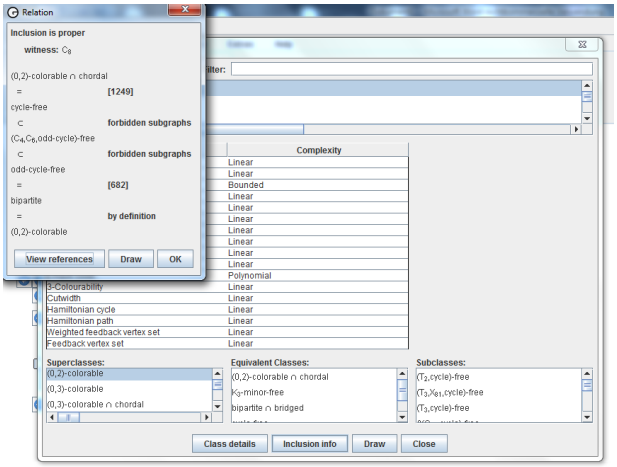
Browse database:

Browse the database by class name. The top half of the dialogue box display a list of classes. The Filter Box can be used to limit the display of classes. Classes are sorted alphabetically, where capitals come before lower case, and special characters are ignored for the sorting order. Clicking on a class will select it and display information in the lower half of the window.
The information displayed is the complexity on the current graph class for one problem, and, below that, super-, sub- and equivalent classes.



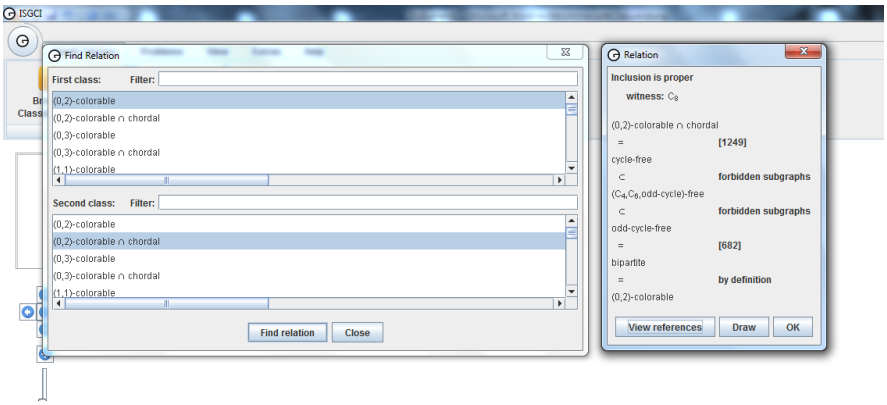
Single clicking a class in one of the three lower lists selects it and double clicking will display information on it in the current window.
Clicking *Class info* will open the Internet page of ISGCI, where further information about the selected Class can be found.

Clicking *Inclusion info* will display the inclusion between the current graph class and the selected one from one of the super/sub/equivalent lists.

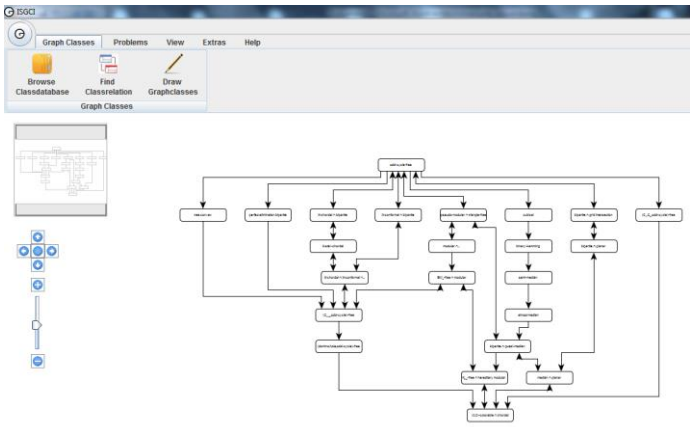


Find relation:

Select a graph class in the top list, and one in the bottom list and press *Find relation* to find the relation between them. The Filter Boxes can be used to limit the display of classes in both lists. If one of the classes is a subclass of the other an inclusion path will be displayed. Otherwise the minimal common superclasses and maximal common subclasses are displayed.

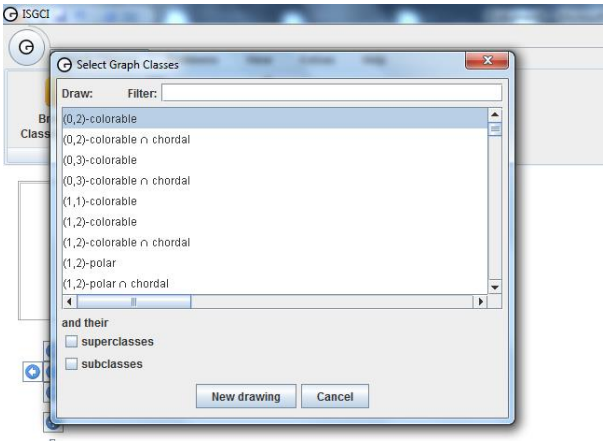


By clicking the Draw Button, the Relation will be drawn as a graph.



Draw Graphclasses:

Select one or more graph classes, whether you would like their super/subclasses to be drawn also, and click *New drawing* to create a new inclusion diagram in the current window. The Filter Box can be used to limit the display of classes.



Problem definitions:

Normal rules for colouring are as follows:

- **Red** for NP-complete classes.
- **Light red** for GI-complete (GraphIsomorphism-complete) classes.
- White for open classes.
- **Dark green** for classes in P.
- **Light green** for classes on which the problem can be solved in linear time.

The colors can be changed.

Some pseudo-problems (cliquewidth, cliquewidth expression) have their own rules; see the descriptions below.

Recognition
: The recognition problem for a graphclass *X* has as input a graph *G* and asks whether *G* is in *X*.

Independent set
: An independent set of a graph is a pairwise independent subset of its vertices. The independent set problem has as input a graph *G* and a natural number *k* and asks whether *G* has an independent set of at least *k* vertices.

Weighted independent set
: The weighted independent set problem has as input a vertex weighted graph *G* and a real number *k* and asks whether *G* has an independent set with total weight at least *k*.

Domination
: A dominating set of a graph G is a subset of its vertices such that every vertex of the graph is either in the set itself or has a neighbour in the set. The domination (or dominating set) problem has as input a graph *G* and a natural number *k* and asks whether *G* has a dominating set of size at most *k*.

Clique
: A clique of a graph is a pairwise adjacent subset of its vertices. The maximum clique problem has as input a graph *G* and a natural number *k* and asks whether *G* has a clique of at least *k* vertices.

Weighted clique
: The weighted clique problem has as input a vertex weighted graph *G* and a real number *k* and asks whether *G* has a clique with total weight at least *k*.

Cliquewidth
: The cliquewidth of a graph is the number of different labels that is needed to construct the graph using the following operations:

  - creation of a vertex with label *i*,
  - disjoint union,
  - renaming labels *i* to label *j*,
  - connecting all vertices with label *i* to all vertices with label *j*.

  The colouring happens in the following way:

  - **Red** for classes whose cliquewidth is not constant bounded.
  - White for open classes.
  - **Light green** for classes with constant bounded cliquewidth.

Cliquewidth expression

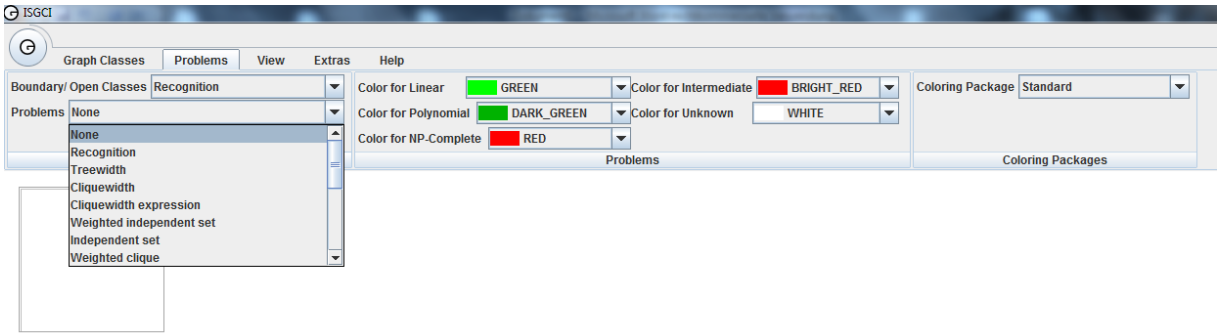
The cliquewidth expression problem has as input a graph *G* and as output an expression constructing that graph that uses only a constant number of labels. The colouring happens in the following way:

- **Red** for classes whose cliquewidth is not constant bounded, or for which finding a clique expression is NP-complete.
- White for open classes.
- **Dark green** for classes that have constant bounded cliquewidth, and a clique expression can be found in polynomial time.
- **Light green** for classes with constant bounded cliquewidth and for which a clique expression can be found in linear time.

Problem menu:

Color for problem:

Selecting a problem from the submenu to this item will color inclusion diagrams according to the complexity of the selected problem, see problem definitions.



Boundary/open classes:

Selecting a problem from the submenu to this item will display a dialogue box with three lists of graph classes:

Minimal NP-complete classes have a direct subclass for which the problem is not NP-complete

,

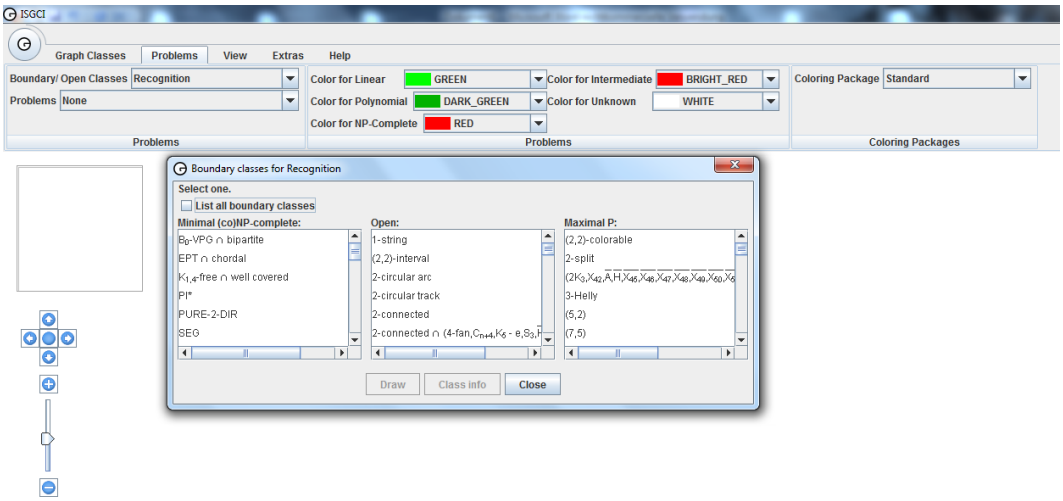
open classes for which the complexity is unknown and

maximal P classes have a direct superclass for which the problem is not known to be in P.

Clicking on a class will select it.

Clicking *Draw* will display the border between P and NP-complete surrounding the selected graph class.

Clicking *Class info* will display information on the selected graph class.

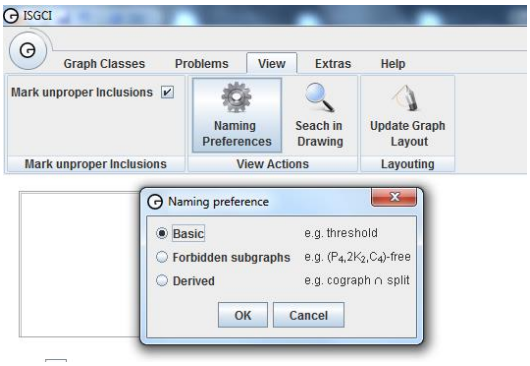


View menu:

Naming preference:

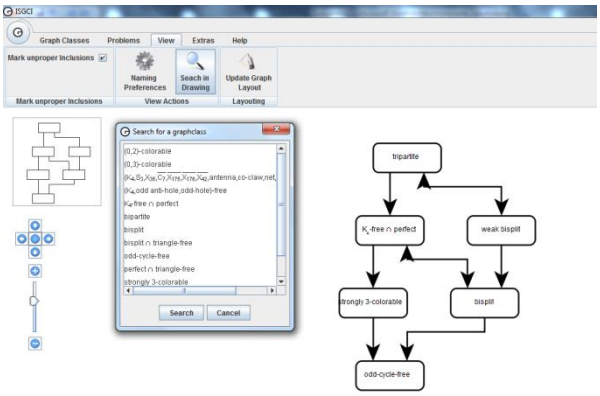
Select the prefered name for classes in inclusion diagrams. A class can be defined in different ways and accordingly have different names. This menu item will popup a dialogue box that allows you to select the prefered name for classes that have more than one name. The options are:

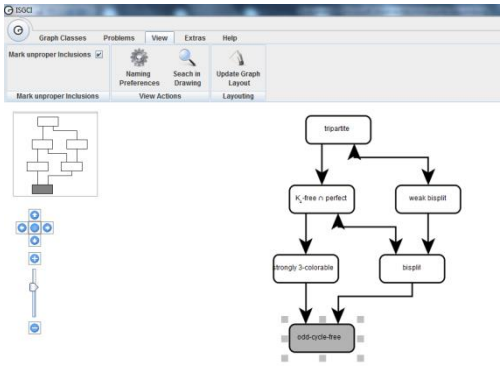
- *Basic*. Any name that does not qualify as one of the other preferences.
- *Forbidden subgraphs*. Characterization by forbidden subgraphs.
- *Derived*. Characterization by set operations (intersect, union, complement) on (an)other graph class(es).



Search in Drawing:

Centers the drawing on a graph class that can be selected from a list of all displayed classes.





Mark unproper inclusions:

When checked, inclusions that are not known to be proper are marked by an arrowhead at the superclass end of the arrow.

Update Graph Layout:

When nodes or edges has been moved, they will get to their old position by pushing this Button.

Extra menu:

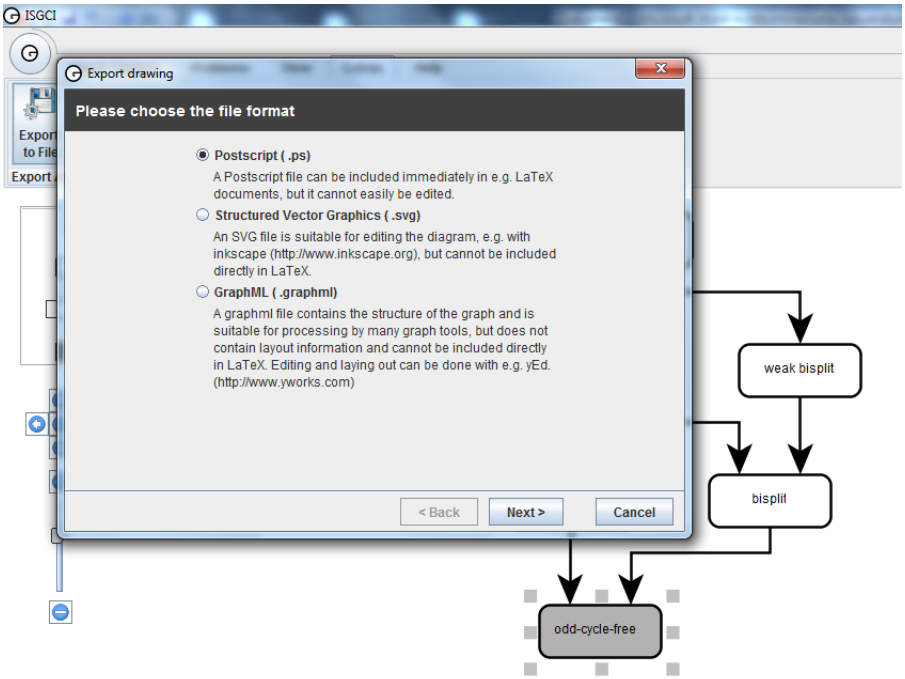
Export to File:

Export the current drawing as an Encapsulated Postscript (EPS), Structured Vector Graphics (SVG) or GraphML file.

A Postscript file can be included directly in your papers. Papersize, scaling and colour or black-and-white print can be selected.

An SVG file cannot be included directly in a LaTeX paper, but it can be manipulated by many drawing programs, allowing you to e.g. add classes.

A GraphML file contains a list of classes and the relations between them, but no drawing.



Hotkeys:

H <- Selects direct neighbors

B <- Selects direct Superclasses

U <- Selects direct Subclasses

Del <- Deletes the selected nodes