

Documentation

Table of contents

1 Working with Concept Explorer.....	2
1.1 Creating a new document.....	2
1.2 Opening existing documents.....	3
1.3 Saving a document.....	3
2 Working on a context.....	3
2.1 Undo/redo support.....	3
2.2 Changing size of a context.....	3
2.3 Compressed view of a context.....	4
2.4 Visualization of the arrow relations.....	4
2.5 Entering data into the context.....	4
2.6 Operations on contexts.....	4
3 Concept Lattices and handling the line diagram.....	5
3.1 Building the concept lattice.....	5
3.2 Interpreting a drawing.....	5
3.3 Visualization modes.....	6
3.4 Changing visualization options	7
3.5 Changing the layout of a lattice.....	8
3.6 Manually adjustment of a drawing.....	9
3.7 Building lattices of subcontexts.....	10
4 Working with implication bases.....	10
4.1 Calculating the Duquenne-Guigues-Basis.....	10
4.2 Searching for associations.....	11
4.3 Performing an attribute exploration.....	11

For downloading ConExp and hint for installation use the menu item Download.

1. Working with Concept Explorer

The ConExp user interface consists of the following parts:

- The **Menu** which is self-explanatory.
- The **Main toolbar** which contains buttons for global application operation as there are:
 - Create a new document;
 - Open;
 - Save;
 - Compute the number of formal concepts;
 - Compute the concept lattice;
 - Perform the attribute exploration;
 - Calculate the Duquenne-Guigues set of implications;
 - Calculate association rules;

additionally we have to mention the combo box which allows to select the update mode for the document components that are computed from the formal context (concept lattice, set of implications and association rules). Currently two update modes are supported. Those are *re-computation* of the affected components or *clear* them. The latter mode is recommended when changing the context reasonably. After realisation of those changes the appropriate operations can be re-obtained for the new context. In the case of many changes the other option would cause a re-computation for every single alteration of the context and may lead to computational expenses if the context is big or dense.

- The **main pane** consists of the following items:
 - **Document-tree**: It displays the structure of the current document and allows to navigate between different views (i.e. context, concept lattice, implications and association rules).
 - **Option-pane**: It allows to edit different options corresponding to the actual view.
 - **View-pane**: It contains the display for the current view. There is a toolbar with specific operations for each view.
- **Statusbar**

1.1. Creating a new document

When starting ConExp a new document is created. Alternatively, one can create a new document by pressing the button *New context* on main toolbar or selecting the menu item *New* in *Files* menu. If a user chooses to open a new document, while working with another that contains unsaved changes, he is asked to save it (or to cancel creating the new document) before creating the new document.

1.2. Opening existing documents

ConExp allows to work with several different data formats. Currently the following formats are supported:

- ConExp native format *.cex: This is an XML-based format. It stores information about the context and the lattice line diagram. In addition to the others it stores the information whether implications and/or association rules were calculated. We recommend to use this format.
- ConImp context data *.cxt: It is possible to work with contexts, that were created using [ConImp](http://www.mathematik.tu-darmstadt.de/~burmeister) (<http://www.mathematik.tu-darmstadt.de/~burmeister>) . The disadvantage is, only the context can be encoded in this format.
- Comma Separated Values *.csv: So far ConExp supports only the import of contexts in this format. Actual separator is semicolon (;). It is assumed, that the first line of the file contains attributes names and the first cell is empty. (I.e, if one has a context with attributes attr1 and attr2, then first line will be the following: ;attr1;attr2.) Each of the succeeding lines should start with an object name followed by a sequence of 0s and 1s. A cross will appearer in all cells of the imported context, where a 1 has been set.
- Object Attribute List *.oal: So far ConExp supports only the import of contexts in this format. Each line contains information on one object starting with the name and followed-up by the possessed attributes. If object obj1 has the attributes attr2 and attr3, the line representing obj1 should look as follows: *obj1:attr2;attr3*.

In addition ConExp allows to reopen documents on which you were working before by selecting one of the items in the sub-menu *Files - Reopen*.

1.3. Saving a document

To save your work, use the menu items *Save* and *Save as* in the menu *Files* or select the button *Save file* on the main toolbar. The recommended storage format is native ConExp format *.cex.

2. Working on a context

2.1. Undo/redo support

ConExp provides undo/redo support for all doings that were performed on the context. One can undo the performed actions by selecting the *Undo last action* button on context editor's toolbar and redo by pressing the *Redo last action* button.

2.2. Changing size of a context

For changing the size of context one should use the properties window on the left hand side of the main frame and enter new numbers for objects/attributes as properties in *object*

count/ attribute count. Moreover, it is possible to add new objects (attributes) into the context by pressing the corresponding button *Add object* or *Add attribute* on context editor's toolbar.

In order to remove some set of objects/attributes, select them in context editor and then perform *Remove object(s)/Remove attribute(s)* action from context editor's context menu.

2.3. Compressed view of a context

Selecting the *Compressed* option on the context editor's property pane, gives a better overview on the context, especially when it is large. Then the width of context's columns will be set just to fit the size of the cross and one can have a better look at the structure of context.

2.4. Visualization of the arrow relations

To visualise arrow relations change the property *show arrow relation* from *don't show* to *show arrow relation*. As we do not want to introduce any theory here, we refer to the [book by Ganter and Wille](#) ([../relatedprojects.html](#)) for the definition and usage of the arrow relations.

2.5. Entering data into the context

Fast editing of contexts: If one needs to input a context of moderate size, one can use the so-called fast context editing.

For this, just use the keys *x* and *.*. A cross or blanc value will be entered accordingly, when staying in an appropriate cell and the cursor will move to the next cell in relation area.

Transformations on selected areas: After marking an area of cells one can transform the content of the incidence relation between objects and attributes. The following transformations are supported:

- **Fill selection**: This fills the selected area of the incidence relation with crosses.
- **Clear selection**: This option clears the content of the selected area.
- **Inverse selection**: Here a cross is replaced by a blanc value and vice-versa.

All these transformations can be performed by using the appropriate command from context menu.

2.6. Operations on contexts

Following operations can be performed on contexts:

- **Object clarification**: This option brings together all objects in the context having equal sets of attributes. The resulting context shows the first occurrence of those rows

only. This operation is executed by the help of the button *Clarify objects* on the context editor's toolbar.

- **Attribute clarification:** This is the analogous operation on the attribute set. It is performed after selecting the *Clarify attributes* button on context editor toolbar.
- **Object set reduction:** Removing all objects from the set of object set that can be obtained as a result of intersection of some other objects is called reduction. In process of performing reduction clarification is also performed. This operation does not change the structure of the concept lattice. Speaking mathematically, the concept lattice of the reduced context is isomorphic to the concept lattice of the original context. This operation is performed by pressing the button *Reduce objects* on the context editor's toolbar.
- **Attribute set reduction:** This is the analogous operation on the set of attribute. It is performed after selecting the *Reduce attributes* button on context editor's toolbar.
- **Context reduction:** Both operations (reduction of the object and attribute set) can be done simultaneously by executing *Reduce context* button on context editor's toolbar.
- **Transposition:** Exchanging the role of objects and attribute set together with the corresponding changes of the relation between them, can be performed by selecting the *Transpose context* button on context editor's toolbar.

3. Concept Lattices and handling the line diagram

3.1. Building the concept lattice

In order to derive the concept lattice from the formal context, use the button *Build Lattice* on the main toolbar. This might be a time consuming process depending on the complexity of the data. The drawing of the lattice appears as a straight line diagram.

Remark: Lattice-layout tools are time consuming. That is why drawings consisting of only one node can appear. After some time the whole structure will follow.

3.2. Interpreting a drawing

ConExp can represent the structure of a finite formal context in form of a concept lattice. This is not only a graph with nodes and edges connecting them, but an ordered structure with a bottom and a top element.

Each node of lattice corresponds to a so called *formal concept*. This is a pair (O, A) , where O is a subset of the object set, A is a subset of the attribute set and some additional properties are satisfied.

The context contains more information on how objects and attributes should be related. From this we can derive the set containing all attributes common to all objects from the set O and only these attributes. For a formal concept this equal to A . Vice versa O contains all objects from context having all attributes of A among their attributes. The set

of objects O is called *extent* of the formal concept (O, A) and set of attributes A is called *intent*. For exact definitions we refer again to the [book by Ganter and Wille](#) ([../relatedprojects.html](#)).

Hence, the intent of the bottom element contains all attributes of the context and its extent is the set of objects having all attributes. This set can be empty, if no such object has been specified. Vice versa contains the extent of the top element of the lattice all objects. The intent of this contains of all attributes common to all objects. If no such attribute has been defined, then this set is empty.

For the drawing of the lattice, we use the so called reduced labelling in order to present information about intents and extents of formal context in a concise way. Each node x of the structure represents a formal concept (O, A) . The extent O of this concept can be received collecting all objects attached to nodes reachable by descending paths from this node x to the bottom element of this lattice.

In the opposite direction works the reading of the intent. The intent A of this concept can be received collecting all attributes attached to nodes reachable by ascending paths from this node x to the top element of this lattice.

If a node is marked by a blue filled upper semi-circle, then there is an attribute attached to this concept. If a node is marked by a black filled lower semi-circle then there is a object attached to this concept. The label of an attributes is always above the node that it is attached to and within a mouse-coloured box. The label of an object is always underneath the node it is attached to within a white box.

Sometimes a node or an edge in a line diagrams is displayed by ConExp in red colour. This means, that the node or edge is placed very close to or even overlapping with some other node. In order to improve the layout, please try manually an adjustment of this layout or use some other layout-tool.

3.3. Visualization modes

Basically, there are two visualization modes, that behave differently, when the drawn lattice does not fit into the existing screen estate. Those are:

- **Scrolling mode:** If the drawn lattice does not fit into the screen estate, the virtual window is enlarged. The user then can see only some part of the lattice drawing. This mode is activated by default.
- **Fit to screen mode:** Here, the drawn lattice is rescaled in order to fit into the available screen estate.

Switching between the two modes can be done by the help of the "Scale picture to fit into the image" button on the lattice visualization pane toolbar. Selecting this button toggles between first mode and second and vice versa.

The following commands make sense only in scrolling mode:

- **Grab and drag:** This command performs panning of the visible area. After pressing this button, the cursor changes to a cross and user can pan the drawing. To switch off this mode, press *Grab and Drag* button once more.
- **Zoom in, Zoom out, No zoom:** These commands perform actions, corresponding to their names.

3.4. Changing visualization options

The following visualization options can be adjusted via *drawing options* properties pane on left part of the screen:

- **Attribs:** Here the user can change to *Show labels*, hence the diagram shows attribute's label at corresponding concept. (See also the preceding remark about reduced labelling.)
- **Objects:** This is the lower label visualization mode. Here the user is given more possibilities to choose from:
 - **Don't show** - no object labels are shown;
 - **Show labels** - shows object labels below the corresponding concepts;
 - **Show own objects** - for concepts, that have some objects attached (has non empty object *contingent*) show number and percentage of objects, that belong exactly (i.e., their attribute set is equal to intent) to this concept;
 - **Show object count** - this shows for every node the exact number (percentage) of objects belonging to the extent of the node's concept;
 - **Stability** - this shows for every node the minimal number of objects that should be removed from context to wipe out the node from concept lattice.
- **Draw node** - this option specifies how the radius of a node is calculated. The possibilities are:
 - **to own objects** - the radius of a node is calculated proportionally to the size of the contingent (the number of objects matching exactly the intent of this node),
 - **fixed radius** - all nodes have a equal radius. The actual node radius is determined by option *Node radius*,
 - **of object extent** - the node radius is calculated proportionally to size of its extent,
 - **stability** - the node radius is calculated proportionally to its stability to dissolve (see description of stability above).
- **Draw edge** - specifies how an edge is drawn. The possible values are:
 - **one pixel** - the edge width is fixed,
 - **no** - edges are not drawn at all,
 - **object** - proportionally to the number of objects, that *pass* through this edge. Equivalent to the option for drawing nodes *of object extent*,
 - **connection** - the size of an edge is proportional to the ratio between extent size of the lower and upper concept that are connected by this edge. This value is equal to confidence of the approximate association rule corresponding to this edge.
- **Highlight** - specifies, which nodes are highlighted, except for selected edges. These

options were created in order to make the exploration of the lattice easier. Possible values of this option are:

- **Filter and ideal** - nodes of the filter (all nodes that are reachable by ascending paths from the selected node to top of the lattice) and the ideal (all nodes that can be reached by descending paths from the selected node to the bottom of the lattices) are highlighted,
 - **Selected** - only the selected node is highlighted,
 - **Neighbors** - the selected node and it's upper and lower neighbors are highlighted,
 - **Ideal** - nodes of the ideal are highlighted,
 - **Filter** - nodes of the filter are highlighted,
 - **No** - no nodes are highlighted. This option may be useful before saving the drawing of the lattice.
- **Label font size** - specifies the size of the font that is used for the upper and lower labels.
 - **Grid size x** - specifies the preferred distance between different nodes on one level of drawing by means of the x coordinate. It is used as a parameter for the lattice layout and changing this value leads to a rescaling of the coordinates of all nodes.
 - **Grid size y** - specifies the preferred distance between nodes on adjacent levels of drawing by means of the y coordinate. Again, it is used as a parameter for the lattice layout and changing this value leads to a rescaling of the coordinates of all nodes.
 - **Node radius** - this parameter specifies the largest radius used for drawing a concept node.

3.5. Changing the layout of a lattice

If the initial drawing of the lattice is not very satisfactorily, then it is recommended to use several different layout algorithms in order to get a first impression and an approximation before starting the manual adjustment of the drawing.

Warning: Performing another layout algorithm destroys the previous drawing. If you do not want to loose work already done, make sure you saved the drawing before you select another algorithm.

The algorithms implemented in ConExp have various options, that can be access through the *Layout options* tab in the properties panel. The following layout algorithms are provided:

- **Minimal intersections** - this is adapted to lattices version of algorithm for laying out hierarchical graphs. It tries to minimize number of intersections between edges. It has no parameters. Usually this algorithm provides best results, but it is pretty slow for the big lattices.
- **Chain decomposition** - it is an adapted version of the algorithm of by M.~Skorsky. The outcome are so called *additive lines diagrams*. It is recommended to use ideal node movement strategy when working with such line diagrams. This algorithm produces very good results for distributive lattices. Here the user can adjust the

following options:

- **Representation** - changes the kind of representation used for formal concepts when calculating its coordinates. The possibilities are attribute-based or object-based calculation.
- **Placement** - determines the assignment of values to the set of vectors. One can choose from three values - *exponential*, *straight* or *angular*.
- **Rotate left, Rotate right** - performs a permutation on the set of vectors - it is used to select the best, if there are several possibilities.

The following two algorithms belong to a family of so-called *force-directed* layout algorithms. They are:

- **Freese layout** - It is an adaptation of the algorithm of Ralph Freese for drawing lattices. It can be found on the [home page of R. Freese](http://www.math.hawaii.edu/~ralph/LatDraw) (<http://www.math.hawaii.edu/~ralph/LatDraw>) . This algorithm has the following parameters:
 - **Attraction** - regulates the attraction force between nodes;
 - **Repulsion** - regulates the repulsion force between nodes;
 - **Angle** - this is not a parameter of the algorithm, but it has a vast influences on the outcome. Freese algorithm performs the layout in 3D space. The angle controls the projection from 3D space into the 2D surface of screen.
- **Force layout** - This force directed algorithm differs in the way the forces are calculated from previous one. The parameters of this algorithm are analogous to the parameters of the previous one.

3.6. Manually adjustment of a drawing

Unfortunately for now there is no layout algorithm producing good results for all lattices. Therefore, the best way to produce good drawing is to perform the manual adjustment of the lattice, when all algorithms failed.

Movements of lattice nodes are constrained in ConExp as far as they maintain a correct parent-child (successor-predecessor) relations between nodes.

The following tools exists in ConExp in order to support manual adjustment of a lattice:

- **Ideal node movement mode** - when moving a node, the whole ideal of the node is moved as well. The switch between this mode and one node movement strategy is performed by pressing the *Toggle node move mode* button.
- **Align nodes to grid** - performs an alignment of the node coordinates to the invisible grid of size 8 on 8 pixels.
- **Storing images of drawing** One of the most frequent uses of ConExp is to produce images of lattice drawings for some future usage. This task can be achieved by creating a good drawing of the lattice and pressing the *Save lattice image* button on lattice visualization pane toolbar afterwards. Currently the formats *png* and *jpeg* are supported for saving images.

3.7. Building lattices of subcontexts

ConExp also provides the ability to build lattices corresponding to a subcontext of the original context. This task can be achieved by using the attribute selection and object selection pane on the right hand side of lattice drawing pane.

After selecting or deselecting names of objects or attributes the new lattice corresponding to the selected subcontext is build. To include all objects (attributes) into this selection use the *Select all objects* (*Select all attributes*) buttons at the bottom of corresponding panes.

Warning: Building lattice of subcontexts leads to a destruction of all information about previous drawing. Please store the image that you obtained if the outcome is useful or you had some work with it.

4. Working with implication bases

4.1. Calculating the Duquenne-Guigues-Basis

The ideas of the following section can be seen as a third approach to the data. Beside the formal context and the lattice diagram one can examine the implications between attributes valid in a context. Again, we ask the user to look up the theory in the recommended literature.

From the Duquenne-Guigues-Basis for implications one can derive all implications valid in a formal context using the Armstrong rules. For calculating this base, one should press the button *Calculate Duquenne-Guigues set of implications* on the main toolbar. The main advantage of Duquenne-Guigues set of implications is that it has minimal size among all possible sets of implications generating all implications that hold in this context.

Implications appearing in the *Implication sets* pane have the following format:

No -Number of objects- Premise -- Conclusion,

where *No* simply means the number of implication in list. *Number of objects* shows, for how many objects this implication holds. *Premise* and *conclusion* are usually lists of attribute names occurring in the premise (conclusion). Even the empty set can be a premise. That means, this implication has the empty premise and therefore the conclusion holds for all objects from context.

Implications are displayed by ConExp in one of the two colours: blue or red. The blue colours indicates that there are objects in context which support this rule. Contrarily, the red colour indicates that there are no objects supporting this implication. Usually such implications show that within the set of objects no element appears having all attributes

of the premise. And indeed, such implications include all attributes from context among premise and conclusion.

4.2. Searching for associations

In contrast to implications, we allow non-strict rules for association rules. If the premise of an association rule holds, the conclusion does not necessarily hold for all objects. However, it is true for some stated percentage of all objects covering the premise of rule. The base of association rules consists of two parts. Of course it includes the base of strict rules (Duquenne-Guigues-Basis) and additionally the base of approximate rules (so called Luxenburger base).

ConExp allows to calculate this base of association rules. To do so, select the button *Calculate association rules* on the main toolbar.

The display format of association rules modifies slightly the format for implications. It is:

No -Number of objects, for which premise holds- Premise =[Rule confidence] -- -Number of objects, for which premise and conclusion holds- Conclusion

In addition to red and blue, as they are used for implications, green indicates the approximate rules.

4.3. Performing an attribute exploration

When calculating the implications for some context, it might turn out that those hold for objects from this context only but not in general for all object from a certain domain of interest. In order to overcome this insufficiency, we suggest to start an *attribute exploration* procedure.

Attribute exploration is an interactive procedure. The program provides implications, that are valid in the given context. These can be seen as question about dependencies between different attributes from some fixed set of attributes. The expert confirms, that such a dependency generally holds by answering *yes*. He rejects this dependency providing a counterexample. If an expert answers correctly on all questions, than at the end of this procedure he receives the set of all implications describing dependencies between different attributes in the domain of interest.

The attribute exploration procedure can either start from an empty context, where only attributes are specified or from a context, where some objects already are described.

To start this procedure, select the button *Start attribute exploration* on the main toolbar.

Then the first implication appears and the user either confirms or rejects it. A third possibility is to stop the attribute exploration procedure. If a user has to reject an implication, a dialogue pane appears and he is asked to provide a counterexample.