\documentclass{article}
\usepackage{fca}

\title{\texttt{fca.sty}}
\subtitle{\LaTeX\-macros for Formal Concept Analysis}
\subtitle{Version 2.1}
\author{Bernhard Ganter}
\author{TU Dresden}
\date{October 1, 2007}

\maketitle

\section*{Abstract}
Formal Concept Analysis is a mathematical field based on the theory of lattices and ordered sets, with applications in data analysis and knowledge processing. To simplify typesetting of FCA-related text, \texttt{fca.sty} provides two environments and some simple macros, just for convenience. \texttt{fca.sty} offers nothing that you could not do without. The two environments are \texttt{cxt} for typesetting small formal contexts as cross-tables, and \texttt{diagram} for making line diagrams of concept lattices. This environment may be of some interest for other purposes as well, since it can also be used for ordered sets and graphs.

A recent version of \texttt{fca.sty} should be available from \url{www.math.tu-dresden.de/~ganter}

\section{Environment \texttt{cxt}}
What this (very simple) environment does can be guessed from an example: The text on the left leads to the output on the right.

\begin{verbatim}
\begin{cxt}
\cxtName{Formula 1}
\att{1.}
\att{2.}
\atr{disqualified}
\obj{x..}{Hamilton}
\obj{.x.}{Alonso}
\obj{.xx}{Massa}
\end{cxt}
\end{verbatim}

\begin{tabular}{|c|c|}
\hline
\textbf{Formula 1} & 1. & 2. & disqualified \\
\hline
Hamilton & $\times$ & & \\
Alonso & & $\times$ & \\
Massa & & $\times$ & $\times$ \\
\hline
\end{tabular}
\texttt{cxt} generates a tabular of the appropriate format. The tabular is defined as soon as the first \texttt{\obj} command is given. Spaces in the preceding lines are not ignored (in this version). Therefore, each line should be ended with a \% . (To be repaired in later versions).

The commands within a \texttt{cxt}-environment are

\texttt{\cxtName{}}  Define the text for the upper left cell of the table. Optional. The default is no text.

\texttt{\att{}}  Give an attribute name. These names are processed in the order in which they are given. Attribute names given after an \texttt{\obj} command are ignored.

\texttt{\atr{}}  Same as \texttt{\att{}}, but with rotated text.

\texttt{\obj{}}  Give an object’s name and its incidence vector, consisting of dots and ‘x’es. The incidences come first, for better alignment. The length of each incidence vector must be the number of attributes.

Each instance of \texttt{\obj} is directly translated to a row of the \texttt{tabular}-environment. It is therefore possible to mix \texttt{\obj} commands with usual \texttt{tabular}-commands.

\texttt{cxt} can handle up to 20 attributes.

The arrow relations may also be used. Instead of \texttt{x} and ., type \texttt{d} (for “down”), \texttt{u} (“up”), or \texttt{b} (“both”), as in the following example:

\begin{verbatim}
\begin{cxt}\
\renewcommand{\cxtArrowStyle}{\footnotesize\color{red}}
\cxtName{Formula 1}\
\att{1.}\
\att{2.}\
\atr{disqualified}\
\obj{xbd}{Hamilton}\
\obj{uxb}{Alonso}\
\obj{bxx}{Massa}\
\end{cxt}
\end{verbatim}

The default for \texttt{\cxtArrowStyle} is \texttt{\footnotesize}. In the above example we have changed it using \texttt{\renewcommand} in order to make the arrows red. The default colour is black.
2 Environment diagram

The diagram environment helps typesetting diagrams of concept lattices, but can be used for ordered sets and graphs as well. Again we start with a small example (for which we have set \unitlength 1.2mm):

\begin{diagram}{40}{55}
\Node{1}{20}{10}
\Node{2}{35}{20}
\Node{3}{5}{30}
\Node{4}{35}{40}
\Node{5}{20}{50}
\Edge{1}{2}
\Edge{1}{3}
\Edge{2}{4}
\Edge{3}{5}
\Edge{4}{5}
%Numbers
\leftAttbox{3}{2}{2}{1.}
\rightAttbox{2}{2}{2}{disqualified}
\rightAttbox{4}{2}{2}{2.}
\leftObjbox{3}{2}{2}{Hamilton}
\rightObjbox{2}{2}{2}{Massa}
\rightObjbox{4}{2}{2}{Alonso}
\end{diagram}

Here are the commands of the diagram–environment:

\begin{diagram}{width}{height} translates to
\begin{picture}(width,height)(\diagramXoffset,\diagramYoffset).

The offsets are zero by default. They can be modified using \renewcommand. Note that the diagram dimensions do not take the labels into account, these may overlap. Putting an \fbox around the above diagram yields (with \unitlength .7mm)
\Node{nodenumber}{xpos}{ypos} Puts a circle at position \((xpos,ypos)\) of the picture. These circles are drawn when \end{diagram} is invoked. The default diameter of the circles is 4 (times \unitlength). It can be changed (for all circles) with \CircleSize{}. The argument must be an integer. The node numbers must be different, consecutive between 0 and 51, but need not necessarily be given in ascending order.

\Numbers Puts numbers inside circles. While working on a diagram it can be helpful to have a picture with numbered nodes. The result of the following command sequence is shown on the right:
\fbox{\unitlength .7mm
\begin{diagram}{40}{55}
\Node{5}{20}{10}
\Node{6}{35}{20}
\Node{4}{5}{30}
\Node{8}{35}{40}
\Node{7}{20}{50}
\Numbers
\end{diagram}}

We recommend to remove the \Numbers--command when the diagram is ready. In most cases it is not a good idea to put text inside the nodes of a diagram.

\Edge{nodenumber1}{nodenumber2} Puts a line between the two nodes with the given numbers. These must have been declared earlier with a \Node--command. For nodes with coordinates \((u,v)\) and \((x,y)\) the command translates to
\[ \text{\fcadrawline}(u,v)(x,y). \]

\fcadrawline\((u,v)(x,y)\) is a pdf\LaTeX{}-compatible reimplementation of the \drawline command, the latter provided by the eepic package.

The \Edge--command is executed immediately. It can be mixed with other picture- and eepic--commands like \spline (see the eepic manual).

\leftAttbox{nodenumber}{xoffset}{yoffset}{text1\ \ text2\ \ ...} This is one of six commands
\[ \{\text{\left, \center, \right}\times\{\text{Attbox, Objbox}\}. \]

These are used to put text to diagram nodes. The Attbox--commands place the text above the corresponding node, the Objbox below. Similarly, the text can be placed to the left, be centered, or be placed to the right of
the labelled node. All this can be modified with the `xoffset, yoffset`-
parameters.

The offsets increase the placement effect. A `\rightObjbox`, which is placed
to the lower right of the corresponding node, will be moved even further
to the lower right if the offsets are positive. Similarly, positive offsets will
push a `\leftAttbox` even more to the upper left, etc.

The text of the label is put in a `\parbox`. It can be broken into several
lines using `\`. The width of the `\parbox` is `\LabelBoxWidth`, with a default
value of 40mm. This can be changed using `\renewcommand`.

The label text and the labelled node are connected with a dotted line. Here
is an example:

```latex
{\unitlength .7mm
\begin{diagram}{40}{15}
  \Node{0}{20}{10}
  \leftAttbox{0}{1}{1}{left\\attribute\ label}
  \rightAttbox{0}{10}{10}{right\attribute\ label}
  \rightObjbox{0}{20}{5}{right\\object\ label}
  \centerObjbox{0}{0}{5}{centered\\object\ label}
\end{diagram}}
```

The style of the labels is given by

```latex
\ObjectLabelStyle Default: \small\baselineskip6pt\rm
\AttributeLabelStyle Default: \small\baselineskip6pt\it.
```

These values can be changed with `\renewcommand`.

This concludes the diagram. The circles representing the
nodes are drawn and filled with white. Everything inside such a circle
(except for the numbers caused by the `\Numbers` command) is erased.

### 2.1 Error messages

Package error messages for the `diagram` environment are not yet implemented.
Errors usually are caused by using node numbers that have not been defined
earlier.
2.2 Fine tuning

You can change certain layout parameters either permanently (by modifying the file fca.sty) or temporarily using the following commands:

\CircleSize{}, Default: 4 \text{ (times } \text{unitlength)}
\NodeColor{}, Default: white
\NodeThickness{}, Default: 1.2pt
\EdgeThickness{}, Default: .8pt
\NoDots,
\renewcommand{\ObjectLabelStyle}{},
\renewcommand{\AttributeLabelStyle}{},
\renewcommand{\LabelBoxWidth}{}.

Except for the first three, these commands can be focussed to single instances, using brackets. For example,

\{\NoDots\centerObjbox{nodenum}{xoffset}{yoffset}{labeltext}\}

generates a single centered object label without dotted line.

2.3 pdfLaTeX compatibility

Version 2 of the \texttt{diagram} environment was designed to be pdfLaTeX compatible. It no longer uses \texttt{eepic.sty}, which is not supported by pdfLaTeX. Instead it uses \texttt{pict2e} and a \texttt{fcadrawline} command, that replaces \texttt{eepic}'s \texttt{drawline}.

2.4 Problems with colour

Since the diagrams are drawn using the \texttt{picture} commands and the \texttt{pict2e} package, we can combine with other packages, for example, with the \texttt{color} package. This allows us to colour edges and text (but not individual nodes, see ??). However, \texttt{color} has a problem with spacing. Changing colors can cause unwanted spaces, and these are particularly unpleasant in pictures. Have a look at the following:

\{
\unitlength 2mm
\begin{diagram}{20}{20}
\Node{1}{5}{5}
\Node{2}{15}{15}
{\color{red}\Edge{1}{2}}
{\color{blue}\Edge{1}{2}}
{\color{red}\Edge{1}{2}}
{\color{blue}\Edge{1}{2}}
\end{diagram}\}
This effect disappears when spaces are avoided. Here is a better version:

\begin{diagram}{20}{20}
\Node{1}{5}{5}
\Node{2}{15}{15}
\Edge{1}{2}
\color{blue}\Edge{1}{2}
\color{red}\Edge{1}{2}
\end{diagram}

2.5 Colouring nodes

Nodes are filled with white by default. This can be changed to any other colour using the command \NodeColor{colorname}. This color then applies to all nodes. colorname must be a color specification as used by the color package. ‘red’, ‘blue’, and ‘green’ should usually work. Other colors may be defined with the \definecolor command, see the documentation of the graphics bundle. For finer colour nuances use the xcolor package and its documentation.

\texttt{\textit{fca.sty}} does not support individual node colouring, but there is a trick to do it nevertheless. Simply include the \texttt{diagram} environment into a \texttt{picture} environment and insert the coloured nodes after the diagram is drawn. How this is done should become clear from the example below. \texttt{\textit{fca.sty}} provides a command \texttt{\ColorNode{colorname}} for this. It overwrites numbers generated by the \texttt{\textit{Numbers}} command.

\begin{verbatim}
{\definecolor{grey}{gray}{.8}
\unitlength 2mm
\NodeThickness{2.5pt}
\EdgeThickness{2.5pt}
\begin{picture}(20,20)
\begin{diagram}{20}{20}
\NodeColor{grey}
\Node{1}{5}{5}
\Node{2}{15}{15}
\Edge{1}{2}
\end{diagram}
\put(5,5){\ColorNode{green}}
\end{picture}}
\end{verbatim}
3 Some macros

For a short description see Figure ??.

\GMI The formal context \((G, M, I)\).
\context The symbol \(\mathbb{K}\), a frequently used name for a formal context.
\context[S] Other letters, such as \(S\), may also be used.
\CL The symbol \(\mathfrak{B}\) for the concept lattice operator. If \(\mathbb{K}\) is a formal context, then \(\mathfrak{B}(\mathbb{K})\) denotes its concept lattice.
\BV same as \CL.
\CLGMI The concept lattice \(\mathfrak{B}(G, M, I)\) of the formal context \((G, M, I)\).
\BVGMI Same as \CLGMI.
\CGMI The set \(\mathfrak{B}(G, M, I)\) of all formal concepts of the formal context \((G, M, I)\).
\BGMI Same as \CGMI.
\extent The extent \(\text{ext}(c)\) of the formal concept \(c := (A, B)\) is \(A\).
\intent The intent \(\text{int}(c)\) of the formal concept \(c := (A, B)\) is \(B\).
\extents The set \(\text{Ext}(\mathbb{K})\) of extents of the formal context \(\mathbb{K}\).
\intents The set \(\text{Int}(\mathbb{K})\) of intents of the formal context \(\mathbb{K}\).
\HNI The subcontext \((H, N, I \cap H \times N)\).
\relI The incidence relation \(I\).
\notI The negation \(\not I\) of the incidence relation.
\bigtimes The product symbol \(\times\).
\DownArrow The \(\searrow\) of the arrow relations.
\Runterpfeil Same as \DownArrow.
\UpArrow The \(\nearrow\) of the arrow relations.
\Hochpfeil Same as \UpArrow.
\DoubleArrow The \(\twoheadrightarrow\) of the arrow relations.
\Doppelpfeil Same as \DoubleArrow.
<table>
<thead>
<tr>
<th>Result</th>
<th>command</th>
<th>German version</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{G}M; I$</td>
<td>$\GMI$</td>
<td>$\text{GMI}$</td>
</tr>
<tr>
<td>$K$</td>
<td>$\text{context}$</td>
<td>$\text{context}$</td>
</tr>
<tr>
<td>$L$</td>
<td>$\text{context}[L]$</td>
<td>$\text{context}[L]$</td>
</tr>
<tr>
<td>$\mathcal{B}$</td>
<td>$\text{CL}$</td>
<td>$\text{CL}$</td>
</tr>
<tr>
<td>$\mathcal{B}(G, M, I)$</td>
<td>$\text{BV}$</td>
<td>$\text{BV}$</td>
</tr>
<tr>
<td>$\mathcal{B}(G, M, I)$</td>
<td>$\text{BGMI}$</td>
<td>$\text{BGMI}$</td>
</tr>
<tr>
<td>$\text{ext}()$</td>
<td>$\text{extent{}$</td>
<td>$\text{extent{}$</td>
</tr>
<tr>
<td>$\text{int}()$</td>
<td>$\text{intent{}$</td>
<td>$\text{intent{}$</td>
</tr>
<tr>
<td>$\text{Ext}()$</td>
<td>$\text{extents{}$</td>
<td>$\text{extents{}$</td>
</tr>
<tr>
<td>$\text{Int}()$</td>
<td>$\text{intents{}$</td>
<td>$\text{intents{}$</td>
</tr>
<tr>
<td>$(H, N, I \cap H \times N)$</td>
<td>$\text{HNI}$</td>
<td>$\text{HNI}$</td>
</tr>
<tr>
<td>$I$</td>
<td>$\text{relI}$</td>
<td>$\text{relI}$</td>
</tr>
<tr>
<td>$I'$</td>
<td>$\text{notI}$</td>
<td>$\text{notI}$</td>
</tr>
<tr>
<td>$\times$</td>
<td>$\text{bigtimes}$</td>
<td>$\text{bigtimes}$</td>
</tr>
<tr>
<td>$\Semi$</td>
<td>$\Downarrow$</td>
<td>$\Downarrow$</td>
</tr>
<tr>
<td>$\Downarrow$</td>
<td>$\text{Runterpfeil}$</td>
<td>$\text{Runterpfeil}$</td>
</tr>
<tr>
<td>$\Uparrow$</td>
<td>$\text{Hochpfeil}$</td>
<td>$\text{Hochpfeil}$</td>
</tr>
<tr>
<td>$\DoubleArrow$</td>
<td>$\text{Doppelpfeil}$</td>
<td>$\text{Doppelpfeil}$</td>
</tr>
<tr>
<td>$\IUpArrow$</td>
<td>$\text{IHochpfeil}$</td>
<td>$\text{IHochpfeil}$</td>
</tr>
<tr>
<td>$\DDArrow$</td>
<td>$\text{DDPfeil}$</td>
<td>$\text{DDPfeil}$</td>
</tr>
<tr>
<td>$\NDDArrow$</td>
<td>$\text{NDDPfeil}$</td>
<td>$\text{NDDPfeil}$</td>
</tr>
</tbody>
</table>

Formal Concept Analysis

Formale Begriffsanalyse

Formalen Begriffsanalyse

Figure 1: Table of \texttt{fca.sty}–macros.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>command</th>
<th>package required</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\vee$</td>
<td>$\text{\texttt{\wedge}}$</td>
<td>stmaryrd</td>
</tr>
<tr>
<td>$\wedge$</td>
<td>$\text{\texttt{\wedge}}$</td>
<td>stmaryrd</td>
</tr>
<tr>
<td>$\bigvee$</td>
<td>$\text{\texttt{\bigvee}}$</td>
<td>stmaryrd</td>
</tr>
<tr>
<td>$\bigwedge$</td>
<td>$\text{\texttt{\bigwedge}}$</td>
<td>stmaryrd</td>
</tr>
<tr>
<td>$\sqcup$</td>
<td>$\text{\texttt{\sqcup}}$</td>
<td>stmaryrd</td>
</tr>
<tr>
<td>$\sqcap$</td>
<td>$\text{\texttt{\sqcap}}$</td>
<td>stmaryrd</td>
</tr>
<tr>
<td>$\bigsqcup$</td>
<td>$\text{\texttt{\bigsqcup}}$</td>
<td>stmaryrd</td>
</tr>
<tr>
<td>$\bigsqcap$</td>
<td>$\text{\texttt{\bigsqcap}}$</td>
<td>stmaryrd</td>
</tr>
</tbody>
</table>

Figure 2: Other symbols that are used in Formal Concept Analysis, and the commands that generate them.
\texttt{\textbackslash IUpArrow} gives \textasciitilde, which is has the same meaning as \textasciitilde, but is drawn in the other direction. This is needed in the definition of \textasciitilde.

\texttt{\textbackslash IHochpfeil} Same as \texttt{\textbackslash IUpArrow}.

\texttt{\textbackslash DDArrow} gives \textasciitilde, the symbol for the transitive closure of the arrow relations.

\texttt{\textbackslash DDPfeil} Same as \texttt{\textbackslash DDArrow}.

\texttt{\textbackslash NDDArrow} gives \textasciitilde, the symbol for the negation of \textasciitilde.

\texttt{\textbackslash NDDPfeil} Same as \texttt{\textbackslash NDDArrow}.

\texttt{\textbackslash Semi} gives \textasciitilde, the symbol for the semi-product.

\texttt{\textbackslash FCA} prints “Formal Concept Analysis”. In most cases, this command does not eat the space following it (thanks to \texttt{\textbackslash xspace}).

\texttt{\textbackslash FBA, \textbackslash FnBA} print “Formale(n) Begriffsanalyse”. These commands also use \texttt{\textbackslash xspace} so that blanks are preserved.

Some symbols that are provided by \LaTeX are listed in Figure ??.

Here is a sample text:

\texttt{\textbackslash FCA offers an elegant way to determine the congruence relations of a complete lattice: The congruence lattice of a doubly founded concept lattice $\texttt{\textbackslash CLGMI}$ is isomorphic to $\texttt{\textbackslash CL(G,M,\textbackslash NDDArrow)}$}.

This translates to:

Formal Concept Analysis offers an elegant way to determine the congruence relations of a complete lattice: The congruence lattice of a doubly founded concept lattice $\mathfrak{R}(G, M, I)$ is isomorphic to $\mathfrak{R}(G, M, \textasciitilde)$.

4 To do

- Improve the placement of the dotted lines connecting nodes with attribute- and object names.
- Allow half-shaded nodes in diagrams, and make them (optionally) automatic for object- and attribute concepts.
- Improve the code to avoid unwanted blanks.