PSTricks Macros for Databases∗

Wolfgang May†

2006/11/26

Abstract
The `pst-dbicons` package provides some useful macros in the database area. It focusses on typesetting ER-Diagrams in a declarative style, i.e., by positioning some nodes and defining the position of all other nodes relative to them by using the standard database terminology. The PSTricks package is required for using `pst-dbicons`, but there is no deep knowledge of PSTricks commands required (although this is useful for exploiting the full functionality).

1 Commands
1.1 ER-Diagrams
ER-Diagrams are a widely used graphical representation formalism for conceptual modeling; especially used in the database community. Their main notions are entities (e.g., persons, cities, or countries), attributes of entities (e.g., name, id-number, age), and relationships between entities (e.g., belongs_to, is_capital_of).

With

\seticonparams{⟨icon-type⟩}{⟨graphics parameters⟩}

the graphical layout of icons (entities, relationships, and attributes) can be specified (by giving the optional argument for PSTricks’ boxes). Default is \texttt{[fillstyle=none]} for all of them. In this documentation, we modify it to

\seticonparams{entity}{shadow=true,fillcolor=lightgray,fillstyle=solid}
\seticonparams{attribute}{fillcolor=lightgray,fillstyle=solid}
\seticonparams{relationship}{shadow=true,fillcolor=lightgray,fillstyle=solid}

From the user’s point of view, every ER-icon (entities, relationships, attributes) has a name which should be typeset in the respective box. Additionally, there are \textit{internal} identifiers of the boxes/nodes to allow for referencing by graphics commands in the standard PSTricks way. These \textit{internal id}’s must start with a letter and must contain only letters and digits. In general, id’s must be unique on every page. Often, the \textit{name} of the entity/relationship/attribute satisfies these conditions; in these cases, the name is also used as id. in other cases, name and id must be specified. Thus, for all commands for typesetting ER-icons, the node id \textit{must} be given, whereas the name to be typeset in the box is optional; if no name is specified, the id is typeset as the name.

∗This file has version number 0.2, last revised 2006/11/26.
†may@informatik.uni-goettingen.de
1people who are familiar with PSTricks know under which conditions non-unique node identifiers can be used.
1.2 Entities and Attributes

With

\entity[(property)]{(id)}{(text)}

an entity type is set as a rectangular node. If the optional argument \textit{(text)} is not given, \textit{(id)} also provides the entity’s text. \textit{(property)} is used for \textit{weak} entity types which are denoted by double lines:

Here is a simple entity \entity[Person] \begin{figure}[h]
\centering
\includegraphics[width=1cm]{Person}
\end{figure}
\textbf{Person} where \textbf{Person} serves also as node text;

\entity[weak]{cty}{City} \begin{figure}[h]
\centering
\includegraphics[width=1cm]{City}
\end{figure}
\textbf{City} makes up a weak entity type where the displayed name is different from the internal name.

\attribute

Attribute icons are set as an oval nodes by

\attribute[(property)]{(id)}{(text)}

Here, the optional argument \textit{(property)} can take the values \texttt{mv} (multivalued; resulting in a double-lined oval) or \texttt{key} (key attribute; resulting in underlining the attribute name). Here are three attributes:

- an ordinary one, \attribute{phone}{phone\_no} \begin{figure}[h]
\centering
\includegraphics[width=1cm]{phone}
\end{figure} ,

- a multivalued attribute, \attribute[mv]{nickname} \begin{figure}[h]
\centering
\includegraphics[width=1cm]{nickname}
\end{figure} ,

- a key attribute, \attribute[key]{pid}{person\_id} \begin{figure}[h]
\centering
\includegraphics[width=1cm]{person}
\end{figure} .

Note that with \texttt{phone\_no}, the optional argument is used for the node text (in case that the node text contains stuff that is not allowed in internal postscript ids – as a rough rule, only characters (and since dbicons 1.14, underscores) are allowed in ids).

\attributeof

In Section 1.3, relationship nodes are defined analogously. At first, it is described how to attach attributes with entities:

\attributeof{(id)}{(dist)}{(angle)}{(property)}{(id2)}{(text)}

typesets an attribute node at angle \textit{(angle)} in distance \textit{(dist)} from the node which is identified by \textit{(id)} (which can be either an entity node or a relationship node). The attribute is made a node named \textit{(id2)}. \textit{(id2)} and \textit{(text)} work as for \entity. The argument \textit{(dist)} is optional, it has not to be given with \textit{every} attribute. By \attrdist{(dist)}, this value can be set to a default (as startup default, 2em is set).
Note that \textit{phone\_no} is set with a bigger distance to \textit{person}. Additionally, the example shows how complex attributes can be set with these commands.

### 1.3 Relationships

With \texttt{\reltype{[\langle property\rangle\{\langle id\}\{\langle text\}\[text\]]}}, a relationship type is set as a diamond-shaped node. Here, the optional argument \langle property\rangle is used to represent \textit{identifying} relationships, used for \textit{weak} entities – thus \langle property\rangle can be equivalently \textit{weak} or \textit{ident} which results in a double-lined relationship type.

For declaratively specifying nodes representing relationships between entities, 

\begin{verbatim}
\relbetween{[\langle property\rangle\{\langle entity-id_1\}\{\langle entity-id_2\}\]]
\\{\langle relationship-id\}\{\langle relationship-name\}\[relationship-name]\[relationship-name]\[relationship-name]\]
\end{verbatim}

is used (which can be augmented with several optional arguments). In the simplest version, as given above, a relationship node is set in-between two entity nodes:

\texttt{\entity{Person} \hspace*{6cm} \entity{Company} \relbetween{works\_at} \{\texttt{works\_at}\}}

Additionally, the \textit{roles} of the entities in the relationship, and the cardinalities can be given (both as independent optional arguments):

\begin{verbatim}
\relbetween{[\langle property\rangle]
\\{\langle entity-id_1\}\{\langle role_1\}\[card_1]\]
\\{\langle entity-id_2\}\{\langle role_2\}\[card_2]\]
\{\langle relationship-id\}\{\langle relationship-name\}\]}\end{verbatim}

Moreover, the placement of the relationship node wrt. the entities can be specified: above, the relationship node was put in the middle of an imaginary line in-between the entity nodes. As a first, small, extension, the placement ratio of the diamond between the entities can be changed (default: 0.5):
The relationship between entities can be represented in a diagram. In the example below, the relationship between `Country` and `City` is shown.

```latex
\entity{Country} \hspace*{6cm} \entity{City}
\relationshipbetween[ident]{Country}[0:n]{City}[1:1]{in}(0.8)
```

If this is still not enough, instead of an imaginary straight line, any other PSTricks node connection command can be used, most likely `\ncarc[...]` with suitable optional arguments:

```latex
\entity{Country} \hspace*{6cm} \entity{City}
\relationshipbetween{Country}[0:n]{City}[1:1]{in}/\ncarc[arcangle=20]/(0.4)
```

For TeX-insiders: Note that the use of `/.../` as argument delimiter avoids collisions with the delimiters used by PSTricks which thus can be used inside `/.../`.

In the above example, although the relationship is placed on an imaginary arc, the connections are straight lines. For these lines, PSTricks commands can be given, too. With this, the full syntax is

```latex
\relationshipbetween[\{property\}]{\{entity-id_1\}...}{\{entity-id_2\}...}{\{relationship-id\} \{relationship-name\}}{\{placement-ratio\}}
```

where all arguments embraced with (...), [...], or /.../ are optional.

With this, an example can be given where two different relationships can hold between a pair of entity types:

```latex
\entity{Country} \hspace*{6cm} \entity{City}
\relationshipbetween{Country}[0:n]{City}[1:1]{in}/\ncarc[arcangle=20]/(0.4)
\relationshipbetween{Country}[1:1]/\ncarc[arcangle=-18]/\{City}[0:1]/\ncarc[arcangle=-12]/\{Capital}/\ncarc[arcangle=-30]/(0.6)
```
Note that for capital, \texttt{ncarc} is used for the lines, where the arcangle of the whole arc is divided suitably to the placement ratio of the diamond node, and that the arcangle of the second entity is negative (since the connection always points from the entity to the relationship).

There is one more interesting special case of binary relationships: recursive relationships. There, the entity node cannot be set “between” the nodes – it has to be placed relative to the entity type which is involved in the relationship.

In this case, the argument \langle connection \rangle has no effect, and last argument \langle placement ratio \rangle has the syntax \langle (distance),(angle) \rangle: the relationship node is set at angle \langle angle \rangle in distance \langle dist \rangle from the entity node. The default for \langle connection \rangle and \langle connection \rangle is set to \texttt{ncarc[arcangleA=10,arcangleB=45]} and \texttt{ncarc[arcangleA=-10,arcangleB=-45]}, respectively.

With the above commands, only binary relationships can be represented. Thus, there is one more macro, relating entity nodes with already existing relationship nodes:

\begin{verbatim}
\inrelationship[\langle property \rangle]{\langle entity-id \rangle}{\langle role \rangle}{\langle card \rangle}
\end{verbatim}

With this, n-ary relations can be represented:

\begin{verbatim}
\begin{tabular}{c}
\entity{Person} \hspace{6cm} \entity{Project} \\
\entity{Machine}
\end{tabular}
\relationshipbetween{Person}{Project}{worksatwith}{works\_at\_with}
\ncarc[arcangle=-20]/
\inrelationship{Machine}{worksatwith}
\end{verbatim}
Additionally, there is a straightforward extension of \inrelationship command to relationship nodes which do not already exist but are set by the command:

\begin{tabular}{c}
\entity{Person} & \entity{Machine} & \entity{Project} \\
\end{tabular}
\begin{tabular}{c}
\entity{Person} \hspace*{6cm} \entity{Project} \hspace{2cm} \entity{Machine} \\
\end{tabular}
\begin{tabular}{c}
\inrelationship{Machine}{worksatwith}[works\_at\_with](4em,90) \\
\inrelationship{Person}{worksatwith} \\
\inrelationship{Project}{worksatwith} \\
\end{tabular}

\begin{flushleft}
1.4 Annotations to Objects
\end{flushleft}

\begin{flushleft}
\note{\langle id \rangle}{\langle text \rangle}{\langle distance \rangle, \langle angle \rangle}
\end{flushleft}

\begin{flushleft}
a node \langle id \rangle can be annotated with a comment \langle text \rangle.
\end{flushleft}

\begin{flushleft}
1.5 Usage
\end{flushleft}

\begin{itemize}
\item Typesetting small fragments of ER-diagrams, e.g., for lecture notes and slides: as shown in the above documentation, fragments of ER-diagrams can easily be integrated with the running text.
\item Typesetting ER-Diagrams for project documentation: here the standard procedure is to design the ER-diagram based on a \texttt{tabular} environment in which the entity nodes are positioned; then, attributes and relationships are positioned in the above declarative way. Using basic PStricks commands, additional entity nodes can also be placed declaratively.
\end{itemize}
There are the following advantages compared to using graphics tools, e.g., xfig:

- the source code of the diagrams is written into the running source code of the document, thus there is no need for a bunch of separate .fig and .eps files.
- more flexibility wrt. renaming and layout changes.
- given the positions of some basic objects, the other objects are set with declarative commands, i.e. they automatically adjust when the positions change.

1.6 Example
1.7 Miscellaneous

With \texttt{\textbackslash nodeconnections\{\texttt{PSTricks-node-connections}\}}, the PSTricks-node connections and label commands do not require any extra horizontal or vertical space.

The command

\begin{verbatim}
\texttt{\textbackslash database\{\langle pos\rangle\}\{\langle width\rangle\}\{\langle height\rangle\}\{\langle graph\_params\rangle\}\{\langle text\rangle\}\{\langle id\rangle\}}
\end{verbatim}

defines a database barrel as a tabular which is vertically centered as given in the optional argument \texttt{\langle pos\rangle} (Default: \texttt{[c]}). \texttt{\langle text\rangle} is the text to be written on the front of the barrel, \texttt{\langle width\rangle} and \texttt{\langle height\rangle} give the width and height of the barrel; both must be a dimension, e.g., 2cm. With the optional graphics parameter \texttt{\langle graph\_params\rangle}, the pstricks parameters \texttt{fillstyle} and \texttt{fillcolor} for the database can be set (see example below). The optional argument \texttt{\langle id\rangle} is used as an internal name, it must start with a letter and must contain only letters and digits. \texttt{\langle id\rangle} is used for \texttt{\rnode} definitions: \texttt{\langle id\rangle} is the whole barrel \texttt{\rnode{lu}{lu}}, \texttt{\rnode{ru}{ru}}, \texttt{\rnode{mu}{mu}}, and \texttt{\rnode{muf}{muf}} denote the points at the upper left corner, upper right corner, upper middle behind and upper middle in front of the barrel. Analogously, \texttt{\rnode{ll}{ll}}, \texttt{\rnode{rl}{rl}}, and \texttt{\rnode{ml}{ml}}, denote the lower left and lower right corner and lower middle in front of the barrel.

\begin{verbatim}
\psset{nodesep=2pt}
\begin{tabular}{lc}
\&\rnode{lu}{lu}\quad\rnode{mu}{mu}\quad\rnode{muf}{muf}\quad\rnode{ru}{ru}\n[1em]
\&\rnode{db}{database}\n[2cm]{\texttt{\textbackslash database}\[t\]{2cm}{1cm}\{}\\texttt{\textbackslash fillstyle=\texttt{solid},\texttt{fillcolor=red}}\{\texttt{\textbackslash database}\}\{\texttt{\example}\}\[5em]\&\rnode{ll}{ll}\quad\rnode{ml}{ml}\quad\rnode{rl}{rl}\n\end{tabular}
\end{verbatim}

2 Code Documentation

Required packages from the PSTricks bundle

1 \texttt{\textbackslash RequirePackage\{pstricks\}}
2 \texttt{\textbackslash RequirePackage\{pst-node\}}
3 \texttt{\textbackslash typeout\{Style \textbackslash basename', Version \textbackslash fileversion\space <\textbackslash filedate>\}}
4 \texttt{\textbackslash ProvidesPackage\{pst-dbicons\}\{\texttt{\filedate} \space\textbackslash space\textbackslash fileversion\}}
\seticonparams \seticonparams{#1}{#2}
5 \def\seticonparams#1#2{\expandafter\def\csname #1@parm\endcsname{[#2]}}
6 \seticonparams{entity}{fillstyle=none}
7 \seticonparams{relationship}{fillstyle=none}
8 \seticonparams{attribute}{fillstyle=none}

\purifylabel Macro \@purifylabel expands the string given in \#2 and stores the result in command given in \#1. Supported TeX commands in string \#2: \_., \textunderscore Example: \@purifylabel{\templabel}{has\_hq\_in}, then use \templabel as nodelabel for pstricks.
9 \def\dbi@purifylabel#1#2{%
10 \begingroup
11 \edef\_\{\string_\%
12 \edef\textunderscore\{\string_\%
13 \edef\x\{\endgroup
14 \def\noexpand#1{#2}%
15 }%
16 \x
17 }

\entity \entity[#1a]{#1b}[#2]
18 \def\entity{%
19 \@ifnextchar[{{\entityi}{\entityi[normal]}}
20 \def\entityi[#1]{%}
21 \@ifnextchar[{{\entityii[#1]}{\entityii[#1][#2]}}
22 \def\entityii#1#2{\csname entity@#1\endcsname{#2}}
23 \def\entity@normal#1#2{%
24 \dbi@purifylabel{\dbi@prfd@nodename}{#1}%
25 \rnode{\dbi@prfd@nodename}{\expandafter\psframebox\entity@parm{\strut#2}}%
26 \def\entity@weak#1#2{%
27 \psset{doubleline=true}%
28 \dbi@purifylabel{\dbi@prfd@nodename}{#1}%
29 \rnode{\dbi@prfd@nodename}{\expandafter\psframebox\entity@parm{\strut#2}}%}
30 \def\attribute{%
31 \@ifnextchar[{{\attributei}{\attributei[sv]}}
32 \def\attributei[#1]{%}
33 \@ifnextchar[{{\attributeii[#1]}{\attributeii[#1][#2]}}
34 \def\attributeii#1#2{\csname attribute@#1\endcsname{#2}}
35 \def\attribute@mv#1#2{\psset{doubleline=true}%
36 \dbi@purifylabel{\dbi@prfd@nodename}{#1}%
37 \expandafter\ovalnode\attribute@parm{\dbi@prfd@nodename}{#2}}%
38 \def\attribute@sv#1#2{%
39 \dbi@purifylabel{\dbi@prfd@nodename}{#1}%
40 \expandafter\ovalnode\attribute@parm{\dbi@prfd@nodename}{#2}}%
41 \def\attribute@key#1#2{%
42 \dbi@purifylabel{\dbi@prfd@nodename}{#1}%
43 \expandafter\ovalnode\attribute@parm{\dbi@prfd@nodename}{\underline{#2}}%}
\attributeof \attributeof{#1}{#2}{#3}{#4}{#5}{#6}
\relationship \relationship[#1a]{#1b}{#2}
\relationshipbetween \relationshipbetween[#1a]{#1b}{#2}{#3}{#4}{#5}{#6}{#7}{#8}{#9}{#10}{#11}{#12}
2.1 Acknowledgements

Thanx to Heiko Oberdiek for adding the handling of ids for internal use in postcsript (used in the \purifylabel macro).
Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols

\@attrdist .......... 44–46
\@eha ............... 178
\@ifnextchar ....... 2, 4, 7, 19, 21, 31, 33, 48,
50, 52, 61, 63, 79, 82,
85, 88, 98, 101, 104,
113, 116, 121, 154, 157,
160, 163, 169, 170, 176
\_ ................... 11
\\ ................... 14
\_ ................... 11

A
\annote .............. 6, 188
\attrdist ........... 2, 46, 48
\attribute .......... 2, 30, 57
\attribute@i .......... 31, 32
\attribute@ii .......... 33, 34
\attribute@key ........ 41
\attribute@mv ........ 35
\attribute@parm .. 37, 40, 43
\attribute@sv ........ 38
\attributeof .......... 2, 44
\attributeof@i ........ 48, 49
\attributeof@ii ....... 50, 51
\attributeof@iii ....... 52–54

B
\basename ............ 3, 177
\begin ................ 13
\begingroup .......... 10

C
\cardpos ......... 6, 75, 76, 135,
138, 148, 151, 173, 186
\csname .......... 5, 22, 34, 64
\dianode .......... 67, 71

D
\database .......... 2, 9
\database@i ........ 2, 3
\database@ii ....... 4–6
\database@iii ...... 7–9
\dbi@linecmd ......... 120, 132, 168, 172, 185
\end .................. 16
\endcsname ........ 5, 22, 34, 64
\ncarc ............ 96, 111
\ncbar ........ 18, 20

E
\edef .......... 11–13
\else ........ 97, 112, 125, 135, 136,
138, 139, 148, 149, 151,
152, 173, 174, 186, 187
\fi .................. 93,
97, 112, 126, 135, 136,
138, 139, 148, 149, 151,
152, 173, 174, 186, 187

F
\ifdbi@recursive ..... 77, 95, 110, 124
\ifx ...... 93, 94, 109, 135, 136,
138, 139, 148, 149, 151,
152, 173, 174, 186, 187
\inrelationship ...... 5, 153
\inrelationship@card 160–162
\inrelationship@linetype ..... 163–165
\inrelationship@newrel .
\inrelationship@newrel@pos
\inrelationship@role
\inrelationship@type 157–159

L
\let ........ 72

N
\ncarc ................ 96, 111
\ncbar ........ 18, 20