The \texttt{pmdraw} package

Matthias Fresacher
matthias.ypg37@slmail.me

January 10, 2024

Abstract
The \texttt{pmdraw} package allows you to draw elements of the diagram monoids, commonly referred to as diagrams. The package provides a lot of flexibility to draw most diagrams and can be customised as needed.

Contents
1 Introduction 2
1.1 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
1.2 Acknowledgments . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
2 Bugs and known problems 2
2.1 Bugs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
2.2 Incompatibilities . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
3 Usage 3
3.1 Bricks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
3.2 Draw commands . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
3.3 Brick options . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
3.3.1 Vertex options . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
3.3.2 Edge options . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
3.3.3 Brace . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
3.3.4 Decorations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
3.3.5 Drawing order . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
3.4 Edge options . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
3.4.1 Non-transversal edge options . . . . . . . . . . . . . . . . . . 19
3.4.2 Transversal edge options . . . . . . . . . . . . . . . . . . . . . . 21
3.5 Brace . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
3.6 Diagram options . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
3.6.1 For all (product) diagrams . . . . . . . . . . . . . . . . . . . . 26
3.6.2 For product diagrams only . . . . . . . . . . . . . . . . . . . . . 29
4 Future features/work 31
5 Examples 32

*This document corresponds to \texttt{pmdraw} v1.2, 2024-01-10.
1 Introduction

1.1 Motivation

Diagram monoids and their closely related algebras are well studied objects with applications in theoretical physics and representation theory in mathematics. The elements of these monoids are called diagrams because they can be drawn graphically as a graph following certain conventions.

My PhD thesis examines properties of these diagram monoids and hence I needed to draw a large number of diagrams in a neat and consistent way. This thesis work formed the basis of this package with only customisability and the manual needed to be added to complete this package.

1.2 Acknowledgments

Most importantly, I thank James East, my PhD supervisor for allowing me to get a little sidetracked in my PhD research and develop and publish this package rather than work on my thesis like I should have.

This package has at its core the basic drawing macro of a diagram as used by James East and Nik Ruškuc in their 2022 publication *Classification of congruences of twisted partition monoids* in Advances in Mathematics. I thank them both for allowing me to build on their work.

I thank everybody that has helped me proof read, spot bugs, provide feedback, suggest improvements or has otherwise contributed to this package. Those that are happy to be named are (in chronological order):

- Nik Ruškuc

This package heavily uses commands that can accept an arbitrary number of arguments as inputs. I thank egreg for their template upon which my commands are based (https://tex.stackexchange.com/a/72915).

As this was my first time using keys, I thank David Carlisle for their very helpful example (https://tex.stackexchange.com/a/542555).

As it turns out, passing my keys to the \draw command of tikz is a little more tricky than one might expect. I thank Andrew Stacey for their solution (https://tex.stackexchange.com/a/64237).

I thank Partha D. for their ready to use out of the box macro for adding grid lines to a tikz drawing (https://tex.stackexchange.com/a/467908).

I also thank everybody on https://tex.stackexchange.com who has answered a questions or solved a problem. There are too many to list you all. Thank you.

2 Bugs and known problems

2.1 Bugs

If you discover a bug, please be so kind and send me a minimum working example of the bug in action and I will work on fixing it.

I would like this package to be as perfect as it can be but I will need your help with this as there simply are too many keys to test every combination or use case.

The following are a list of currently known bugs that will be addressed:
1. When the radius of the arc for non-transversal edges is too large compared to the distance between the two nodes, the arcs overlap. The current solution is to manually specify a \texttt{level} or \texttt{height} that makes the arcs smaller.

2. Baseline adjustments are hard coded so when changing \texttt{scale} or text size, for example, the vertical position of diagrams is likely to be incorrect. The current solution is to manually adjust the \texttt{baseline} within the \texttt{tikz} key.

2.2 Incompatibilities

The following are a list of known problems or incompatibilities that are not scheduled to be resolved anytime soon:

1. None at the moment

3 Usage

To use the package, one must understand that all diagrams are created from a basic building block called a brick.

3.1 Bricks

\texttt{brick} A brick is a diagram that contains no dots. For example,

\begin{center}
\begin{tikzpicture}
\node at (0,0) [shape=diamond] (A) {};
\node at (1,0) [shape=diamond] (B) {};
\node at (2,0) [shape=diamond] (C) {};
\node at (3,0) [shape=diamond] (D) {};
\draw (A) -- (B) -- (C) -- (D);
\end{tikzpicture}
\end{center}

is a brick whilst

\begin{center}
\begin{tikzpicture}
\node at (0,0) [shape=diamond] (A) {};
\node at (1,0) [shape=diamond] (B) {};
\node at (2,0) [shape=diamond] (C) {};
\node at (3,0) [shape=diamond] (D) {};
\draw (A) -- (B) -- (C) -- (D);
\end{tikzpicture}
\end{center}

<brick> is not a brick. The syntax for a brick contains four components and is as follows:

\begin{verbatim}
<brick> = [options]<Uedges>*<Ledges>*<Tedges>
\end{verbatim}

brick/<options> where the \texttt{options} are described in Section 3.3.

\begin{itemize}
\item \texttt{Uedges} is a list of upper non-transversal edges. An upper non-transversal edges has three components \texttt{Uedges} = [options]<Svertex>*<Tvertex>
\item \texttt{Ledges} is a list of lower non-transversal edges. A lower non-transversal edges has three components \texttt{Ledges} = [options]<Svertex>*<Tvertex>
\end{itemize}

\texttt{Svertex} is the starting vertex and \texttt{Tvertex} is the terminating vertex of the edge. Both \texttt{Svertex} and \texttt{Tvertex} are a $x$-coordinate and it is assumed that \texttt{Svertex} < \texttt{Tvertex}.

\texttt{level} A list of upper non-transversal edges is simply each upper non-transversal edge written after the other. For example a list with one upper non-transversal edges is

\{1\}{2}\}

whilst one with three upper non-transversal edges may be

\{1\}{2}\{level=1\}{3}\{4\}{5}\{6\}
For ease of reading this will coded as

\{1\}{2}
\{level=1\}\{3\}\{4\}
\{5\}\{6\}

\pmdEmpty

Should a list be empty, the command \pmdEmpty should be used.

\pmdBrick[ % Options
degree=5
]{ % Upper non-transversal edges
  \{1\}\{2\}
  \{2\}\{5\}
}{ % Lower non-transversal edges
  \{1\}\{2\}
  \{3\}\{4\}
  \{4\}\{5\}
}{ % Transversal edges
  \pmdEmpty
}

3.2 Draw commands

This package has three draw commands.

\pmdBrick draws a brick and has the syntax \pmdBrick<brick>.

\pmdBrick[ % Options
degree=5
]{ % Upper non-transversal edges
  \{1\}\{2\}
  \{2\}\{5\}
}{ % Lower non-transversal edges
  \{1\}\{2\}
  \{3\}\{4\}
  \{4\}\{5\}
}{ % Transversal edges
  \pmdEmpty
}

\pmdDiagram draws arbitrarily many bricks separated by dots and has the syntax

\pmdDiagram[^options]{<bricks>}

diagram[^options> where the ^options> are described in Section 3.6 and ^bricks> is a list of
^bricks> ^brick>-s, with each brick enclosed in curly brackets.
It should be noted that \texttt{pmdBrick} is simply a wrapper for a single brick that calls \texttt{pmdDiagram} with no diagram options. This means that, for example, the \texttt{grid} key is not available for \texttt{pmdBrick} and \texttt{pmdDiagram} must be used instead to access the \texttt{grid} key.

For edges that go across multiple bricks, know that the coordinate positions of the vertices extends infinitely in both directions. If using default settings, the dots between bricks occupies one vertex position. To help with counting and identifying the correct coordinate position, the \texttt{grid} flag may also be used, see Section 3.6.1.
\pmdDiagram{
  \{ % Brick 1
    \[ % Options
    \text{degree=3,}
    \text{brace=}
    \{
      \text{left=2,}
      \text{right=6,}
      \text{label=\{n\}}
    \}
    \}
  \}{ % Upper non-transversal edges
    \pmdEmpty
  \}{ % Lower non-transversal edges
    \pmdEmpty
  \}{ % Transversal edges
    \{1}\{1\}
    \{2\}\{6\}
    \{3\}\{6\}
  \}
  \}{ % Brick 2
    \[ % Options
    \text{degree=2}
    \]
  \}{ % Upper non-transversal edges
    \pmdEmpty
  \}{ % Lower non-transversal edges
    \pmdEmpty
  \}{ % Transversal edges
    \{1\}\{-1\}
    \{2\}\{-2\}
  \}
}\}

\pmdProduct\quad\text{draws two diagrams on top of each other as is the case during multiplication or concatenation and has the syntax}

\pmdProduct[<options>]{<Aedges>}{<Tbricks>}{<Bbricks>}

\text{where the <options> are described in Section 3.6.}

\text{<Aedges> is a list of edges that are added during the concatenation process. An added edge has two components, namely <Aedges> = \{<Svertex>\}<Tvertex>, where <Svertex> is the starting vertex and <Tvertex> is the terminating vertex of the collection of added edges. Both <Svertex> and <Tvertex> are a x-coordinate and it is assumed that <Svertex> < <Tvertex>.}

\text{<Tbricks> is a list of bricks that correspond to the top diagram, with each brick enclosed in curly brackets.}

\text{<Bbricks> is a list of bricks that correspond to the bottom diagram, with each brick enclosed in curly brackets.}
\pmdProduct{ % Added edges
  \{1\}\{5\}
  \{7\}\{8\}
}{ % Top diagram
  { % Brick 1
    [ % Options
      degree=5
    ]{ % Upper non-transversal edges
      \pmdEmpty
    }{ % Lower non-transversal edges
      \pmdEmpty
    }{ % Transversal edges
      \{1\}\{1\}
      \{2\}\{2\}
      \{3\}\{3\}
      \{4\}\{4\}
      \{5\}\{5\}
    }
  }{ % Brick 2
    [ % Options
      degree=2
    ]{ % Upper non-transversal edges
      \pmdEmpty
    }{ % Lower non-transversal edges
      \pmdEmpty
    }{ % Transversal edges
      \{1\}\{1\}
      \{2\}\{2\}
    }
  }
}{ % Bottom diagram
  { % Brick 1
    [ % Options
      degree=5
    ]{ % Upper non-transversal edges
      \{2\}\{3\}
      \{4\}\{5\}
    }{ % Lower non-transversal edges
      \{2\}\{3\}
      \{4\}\{5\}
    }{ % Transversal edges
      \{1\}\{1\}
    }
  }{ % Brick 2
    [ % Options
      degree=2
    ]{ % Upper non-transversal edges
      \{1\}\{2\}
    }{ % Lower non-transversal edges
      \{1\}\{2\}
    }{ % Transversal edges
      \pmdEmpty
    }
  }
}
3.3 Brick options

The following are all of the keys that are available for bricks.

3.3.1 Vertex options

degree specifies the degree, that is, the number of vertices in the top and bottom row of the diagram. Must be a positive integer and must be provided unless using degree top and degree bottom.

\pmdBrick[ % Options
degree=5
 ]{ % Upper non-transversal edges
  (1){2}
  (2){5}
}{ % Lower non-transversal edges
  (1){2}
  (3){4}
}{ % Transversal edges
  (1){1}
  (3){4}
}

degree top specifies the degree, that is, the number of vertices in the top (bottom) row of the diagram. Must be a positive integer and must be provided as a pair degree top and degree bottom.

\pmdBrick[ % Options
degree top=5,
degree bottom=4
 ]{ % Upper non-transversal edges
  (1){2}
  (2){5}
}{ % Lower non-transversal edges
  (1){2}
  (3){4}
}{ % Transversal edges
  (1){1}
  (3){4}
}

blank top specifies the number of blank vertices to the left of the top (bottom) row of the diagram. Default is zero.
\pmdBrick[
  % Options
  blank top=1,
  degree top=4,
  degree bottom=5
]{
  % Upper non-transversal edges
  {2}{5}
}{
  % Lower non-transversal edges
  {1}{2}
  {3}{4}
}{
  % Transversal edges
  {3}{4}
}

It should also be noted that the blank top and blank bottom can also be used to create more space between bricks and the dots between bricks by setting a large number. Likewise if set to \text{-}1 and the dots is redefined to not draw, the space between bricks can be removed altogether. See Examples 13-15.

row sep specifies the vertical separation between rows of vertices. Default is two.

\pmdBrick[
  % Options
  degree=5,
  row sep=4
]{
  % Upper non-transversal edges
  {1}{2}
  {2}{5}
}{
  % Lower non-transversal edges
  {1}{2}
  {3}{4}
}{
  % Transversal edges
  {1}{1}
  {3}{4}
}

vertices passes through options to the \draw command of vertices. The default is no argument.

\pmdBrick[
  % Options
  degree=5,
  vertices=blue
]{
  % Upper non-transversal edges
  {1}{2}
  {2}{5}
}{
  % Lower non-transversal edges
  {1}{2}
  {3}{4}
}{
  % Transversal edges
  {1}{1}
  {3}{4}
}

vertices options allows for the redefinition of the command that draws the vertices. It has two arguments, the x and y-position of the vertex. The default is

\fill[apply style/.expand once=\pmdraw@drawVertexOptions] (#1,#2) circle (.17);

An example use is
\pmdBrick[ % Options
degree=5,
vertices options={
  \draw[very thick] (#1,#2) circle (.25);
}
]{ % Upper non-transversal edges
  {1}{2}
  {2}{5}
}{ % Lower non-transversal edges
  {1}{2}
  {3}{4}
}{ % Transversal edges
  {1}{1}
  {3}{4}
}

flags that allow for the drawing of labels for the vertices. Use labels for labels on both the top and bottom or set individually. The default is no labels.

\pmdBrick[ % Options
degree=5,
labels bottom
]{ % Upper non-transversal edges
  {1}{2}
  {2}{5}
}{ % Lower non-transversal edges
  {1}{2}
  {3}{4}
}{ % Transversal edges
  {1}{1}
  {3}{4}
}

sets the starting value of the labels for the vertices. The default is one.

\pmdBrick[ % Options
degree=5,
labels, labels top start=3,
labels bottom start=-2
]{ % Upper non-transversal edges
  {1}{2}
  {2}{5}
}{ % Lower non-transversal edges
  {1}{2}
  {3}{4}
}{ % Transversal edges
  {1}{1}
  {3}{4}
}

passes through options to the \draw and node command of the labels of vertices. The default is no argument.

\pmdBrick[ % Options
degree=5,
labels top draw
labels top node
]{ % Upper non-transversal edges
  {1}{2}
  {2}{5}
}{ % Lower non-transversal edges
  {1}{2}
  {3}{4}
}{ % Transversal edges
  {1}{1}
  {3}{4}
}
\pmdBrick[ % Options
degree=5,
labels,
labels top node=red,
labels bottom node={right=2pt}
]{ % Upper non-transversal edges
{1}{2}
{2}{5}
}{ % Lower non-transversal edges
{1}{2}
{3}{4}
}{ % Transversal edges
{1}{1}
{3}{4}
}

\begin{center}
\begin{tikzpicture}
\node (1) at (0,0) [circle,fill] {1};
\node (2) at (1,0) [circle,fill] {2};
\node (3) at (2,0) [circle,fill] {3};
\node (4) at (3,0) [circle,fill] {4};
\node (5) at (4,0) [circle,fill] {5};

\draw (1) -- (2);
\draw (2) -- (5);
\draw (1) -- (3);
\draw (3) -- (4);
\draw (3) -- (1);
\draw (4) -- (1);
\draw (5) -- (1);
\end{tikzpicture}
\end{center}

labels top options allows for the redefinition of the command that draws the labels of the vertices.

labels bottom options They have three arguments, the $x$ and $y$-position of the vertex and the label value.

The default for the top is
\[
\text{\texttt{\draw[apply style/.expand once=\pmdraw@drawLabelTDrawOptions] (#1,#2) node[above=6pt,apply style/.expand once=\pmdraw@drawLabelTNodeOptions] {(#3)};}}
\]
and for the bottom
\[
\text{\texttt{\draw[apply style/.expand once=\pmdraw@drawLabelBDrawOptions] (#1,#2) node[below=6pt,apply style/.expand once=\pmdraw@drawLabelBNodeOptions] {(#3')};}}
\]

An example use is
\[
\text{\texttt{% Options
degree=5,
labels,
labels bottom options={
\draw (#1,#2) node[below=6pt,blue] {(#3'')};
}}}
\]

\begin{center}
\begin{tikzpicture}
\node (1) at (0,0) [circle,fill] {1}'';
\node (2) at (1,0) [circle,fill] {2}'';
\node (3) at (2,0) [circle,fill] {3}'';
\node (4) at (3,0) [circle,fill] {4}'';
\node (5) at (4,0) [circle,fill] {5}'';

\draw (1) -- (2);
\draw (2) -- (5);
\draw (1) -- (3);
\draw (3) -- (4);
\draw (3) -- (1);
\draw (4) -- (1);
\draw (5) -- (1);
\end{tikzpicture}
\end{center}
The `no dots` flag removes the drawing of dots to the left of the current brick in diagrams. Default is to draw dots except for the first brick.

```
pmdDiagram{
  { % Brick 1
    [ % Options
      degree=2,
      brace={
        left=1,
        right=4,
        label={n}
      }
    ]
    %{ % Upper non-transversal edges
      pmdeempty
    }{% % Lower non-transversal edges
      pmdeempty
    }{% % Transversal edges
      {1}{1}
      {2}{2}
    }
  }
}{ % Brick 2
  [ % Options
    degree=1,
    no dots
  ]% % Upper non-transversal edges
  pmdeempty
}{ % Lower non-transversal edges
  pmdeempty
}{ % Transversal edges
  {1}{1}
}
}
```

dots draw passes through options to the \texttt{draw} and \texttt{node} command of the dots. The \texttt{dots node} default is no argument.
3.3.2 Edge options

These options should be read in conjunction with Section 3.4. 
levels specifies how many horizontal levels there are for non-transversal edges. Default is one.
\pmdBrick[ % Options
  degree=5,
  levels=3 ]{ % Upper non-transversal edges
   [level=3]{{1}{5}}
   [level=2]{{1}{4}}
   [level=1]{{2}{3}}
 }{ % Lower non-transversal edges
   [level=3]{{1}{5}}
   [level=2]{{1}{4}}
   [level=1]{{2}{3}}
 }{ % Transversal edges
  \pmdEmpty
 }{ % Options
  degree=5,
  levels top=3,
  levels bottom=2
 }{ % Upper non-transversal edges
   [level=3]{{1}{5}}
   [level=2]{{1}{4}}
   [level=1]{{2}{3}}
 }{ % Lower non-transversal edges
   [level=2]{{1}{4}}
   [level=1]{{2}{3}}
 }{ % Transversal edges
  \pmdEmpty
 }

\pmdBrick[ % Options
  degree=5,
  levels=3,
  levels sep=0.3 ]{ % Upper non-transversal edges
   [level=3]{{1}{5}}
   [level=2]{{1}{4}}
   [level=1]{{2}{3}}
 }{ % Lower non-transversal edges
   [level=3]{{1}{5}}
   [level=2]{{1}{4}}
   [level=1]{{2}{3}}
 }{ % Transversal edges
  \pmdEmpty
 }

\pmdBrick[ % Options
  degree=5,
  levels top=3,
  levels bottom=2
 ]{ % Upper non-transversal edges
   [level=3]{{1}{5}}
   [level=2]{{1}{4}}
   [level=1]{{2}{3}}
 }{ % Lower non-transversal edges
   [level=3]{{1}{5}}
   [level=2]{{1}{4}}
   [level=1]{{2}{3}}
 }{ % Transversal edges
  \pmdEmpty
 }

\pmdBrick[ % Options
  degree=5,
  levels=3,
  levels sep top=3,
  levels sep bottom=2
 ]{ % Upper non-transversal edges
   [level=3]{{1}{5}}
   [level=2]{{1}{4}}
   [level=1]{{2}{3}}
 }{ % Lower non-transversal edges
   [level=3]{{1}{5}}
   [level=2]{{1}{4}}
   [level=1]{{2}{3}}
 }{ % Transversal edges
  \pmdEmpty
 }

\pmdBrick[ % Options
  degree=5,
  levels=3,
  levels sep=0.3
 ]{ % Upper non-transversal edges
   [level=3]{{1}{5}}
   [level=2]{{1}{4}}
   [level=1]{{2}{3}}
 }{ % Lower non-transversal edges
   [level=3]{{1}{5}}
   [level=2]{{1}{4}}
   [level=1]{{2}{3}}
 }{ % Transversal edges
  \pmdEmpty
 }

\pmdBrick[ % Options
  degree=5,
  levels=3,
  levels sep top=3,
  levels sep bottom=2
 ]{ % Upper non-transversal edges
   [level=3]{{1}{5}}
   [level=2]{{1}{4}}
   [level=1]{{2}{3}}
 }{ % Lower non-transversal edges
   [level=3]{{1}{5}}
   [level=2]{{1}{4}}
   [level=1]{{2}{3}}
 }{ % Transversal edges
  \pmdEmpty
 }

levels top specifies how many horizontal levels there are for upper (lower) non-transversal edges. Default is one.

levels sep specifies the vertical separation between horizontal levels for non-transversal edges. Default is evenly spaced within the space available for the given number of levels.

levels sep top specifies the vertical separation between horizontal levels for upper (lower) non-transversal edges. Default is evenly spaced within the space available for the given number of levels.
\begin{verbatim}
\pmdBrick[ % Options
dergee=5,
levels=3,
levels sep top=0.4,
levels sep bottom=0.15
]{ % Upper non-transversal edges
[level=3]{1}{5}
[level=2]{1}{4}
[level=1]{2}{3}
}{ % Lower non-transversal edges
[level=3]{1}{5}
[level=2]{1}{4}
[level=1]{2}{3}
}{ % Transversal edges
\pmdEmpty
}
\end{verbatim}

The default is no argument.

\begin{verbatim}
\pmdBrick[ % Options
dergee=5,
edges upper=blue,
edges lower={red, dotted, very thick},
edges transversal=dashed
]{ % Upper non-transversal edges
(1){2}
(2){5}
}{ % Lower non-transversal edges
(1){2}
(3){4}
}{ % Transversal edges
(1){1}
(3){4}
}
\end{verbatim}

The flag ensures edges are drawn before vertices. The default is vertices are drawn first.

\begin{verbatim}
\pmdBrick[ % Options
dergee=5,
edges lower={red, very thick},
edges first
]{ % Upper non-transversal edges
(1){2}
(2){5}
}{ % Lower non-transversal edges
(1){2}
(3){4}
}{ % Transversal edges
(1){1}
(3){4}
}
\end{verbatim}
The `transversals first` flag ensures transversal edges are drawn before non-transversal edges. The default is non-transversal edges are drawn first.

```latex
\pmdBrick[ % Options
degree=5,
edges lower={red, very thick},
transversals first
]{ % Upper non-transversal edges
  (1){2}
  (2){5}
}{ % Lower non-transversal edges
  (1){2}
  (3){4}
}{ % Transversal edges
  (1){1}
  (3){4}
}
```

3.3.3 Brace

The `brace` specifies instructions for the drawing of a brace with the brace keys. These are described in Section 3.5. The default is no brace is drawn.

```latex
\pmdDiagram{
  { % Brick 1
    [ % Options
degree=2,
brace={
      left=1,
      right=4,
      label={n}
    }
  }
  { % Upper non-transversal edges
    \pmdEmpty
  }
  { % Lower non-transversal edges
    \pmdEmpty
  }
  { % Transversal edges
    (1){1}
    (2){2}
  }
}

{ % Brick 2
  [ % Options
degree=1
  }
  { % Upper non-transversal edges
    \pmdEmpty
  }
  { % Lower non-transversal edges
    \pmdEmpty
  }
  { % Transversal edges
    (1){1}
  }
}
```
brace draw passes through options to the `\draw` and `node` command of the brace. Must brace node be given before the `brace` key. The default is no argument.

```
\pmdDiagram{
  { % Brick 1
    [ % Options
      degree=2,
      brace draw=thin,
      brace node={pos=0.25},
      brace={
        left=1,
        right=4,
        label={n}
      }
    }
    % Upper non-transversal edges
    \pmdEmpty
  }
  % Lower non-transversal edges
  \pmdEmpty
  % Transversal edges
  {1}{1}
  {2}{2}
}
}
{ % Brick 2
  [ % Options
    degree=1
  ]
  % Upper non-transversal edges
  \pmdEmpty
  % Lower non-transversal edges
  \pmdEmpty
  % Transversal edges
  {1}{1}
}
}
```

3.3.4 Decorations

decorate before provides a hook to draw additional `tikz` elements before and after the brick is decorate after drawn. The default is empty.
These hooks can also be used for more sophisticated manipulation of diagrams by inserting not just `tikz` elements but arbitrary code before and after draw a brick.

\pmdDiagram{
  \pmdBrick[ % Options
    degree=4,
    decorate after={
      \draw[
        very thick, red,
        rounded corners=5pt
      ] (0.6, -0.4) rectangle (1.4, 2.4);
    }
  ]{ % Upper non-transversal edges
    {1}{2}
    {2}{3}
  }{ % Lower non-transversal edges
    {2}{3}
    {3}{4}
  }{ % Transversal edges
    {1}{1}
    {3}{4}
  }
}

\pmdDiagram{
  \pmdBrick[ % Options
    degree=3,
    decorate before={\begin{scope}[red,shift={(1,1)},rotate=30]},
    decorate after={\end{scope}}
  ]{ % Upper non-transversal edges
    \pmdEmpty
  }{ % Lower non-transversal edges
    \pmdEmpty
  }{ % Transversal edges
    {1}{2}
    {2}{3}
    {3}{1}
  }
}
3.3.5 Drawing order

For clarity, this is the default drawing order for a brick:

1. decorate before
2. Top row of vertices
3. Bottom row of vertices
4. Upper non-transversal edges
5. Lower non-transversal edges
6. Transversal edges
7. decorate after

This order can be altered using the edges first and transversals first flags.

For the drawing order of a diagram and product diagram, see the decorate before and decorate after keys in Section 3.6.1.

3.4 Edge options

3.4.1 Non-transversal edge options

level specifies the horizontal level of a non-transversal edges. Default is one.

\pmdBrick[ % Options
degree=5,
levels=3
]{ % Upper non-transversal edges
 [level=3]{1}{5}
 [level=2]{1}{4}
 [level=1]{2}{3}
}{ % Lower non-transversal edges
 [level=3]{1}{5}
 [level=2]{1}{4}
 [level=1]{2}{3}
}{ % Transversal edges
 \pmdEmpty

height specifies a manually set horizontal height of a non-transversal edges. Default is to use levels.
allows for the redefinition of the command that draws the non-transversal edge. It has two arguments, the start and end $x$-position of the edge. The default for an upper non-transversal is

\draw[apply style/.expand once=\pmdraw@drawUedgesDrawOptions] (#1,\pmdraw@rowSep) arc (180:270:\pmdraw{edgeHeight}); % Draw left arch
\draw[apply style/.expand once=\pmdraw@drawUedgesDrawOptions] (#1+\pmdraw{edgeHeight},\pmdraw@rowSep-\pmdraw{edgeHeight}) -- (#2-\pmdraw{edgeHeight},\pmdraw@rowSep-\pmdraw{edgeHeight}); % Draw straight line
\draw[apply style/.expand once=\pmdraw@drawUedgesDrawOptions] (#2-\pmdraw{edgeHeight},\pmdraw@rowSep-\pmdraw{edgeHeight}) arc (270:360:\pmdraw{edgeHeight}); % Draw right arc

and for a lower non-transversal

\draw[apply style/.expand once=\pmdraw@drawLedgesDrawOptions] (#1,0) arc (180:90:\pmdraw{edgeHeight}); % Draw left arch
\draw[apply style/.expand once=\pmdraw@drawLedgesDrawOptions] (#1+\pmdraw{edgeHeight},\pmdraw{edgeHeight}) -- (#2-\pmdraw{edgeHeight},\pmdraw{edgeHeight}); % Draw straight line
\draw[apply style/.expand once=\pmdraw@drawLedgesDrawOptions] (#2-\pmdraw{edgeHeight},\pmdraw{edgeHeight}) arc (90:0:\pmdraw{edgeHeight}); % Draw right arc

An example use is

\pmdBrick[ % Options
degree=5 ]{ % Upper non-transversal edges
 {1}{2} }{ % Lower non-transversal edges
 {options={
 \draw[blue, very thick] (#1,0) -- (#2,0); 
 }}{1}{2}
 {3}{4} }{ % Transversal edges
 {1}{1} 
 {3}{4} 
}
3.4.2 Transversal edge options

The `height` option draws transversal edges with a horizontal component and specifies the manually set horizontal height of the transversal edges. Default is to use non-horizontal transversal edges.

```latex
\pmdDiagram{
  { % Brick 1
    [ % Options
      degree top=5,
      degree bottom=6
    ]{ % Upper non-transversal edges
      {2}{3}
      {4}{5}
    }{ % Lower non-transversal edges
      {3}{4}
      {5}{6}
    }{ % Transversal edges
      [height=1]{10}{2}
      {1}{1}
    }
  }
  { % Brick 2
    [ % Options
      degree top=3,
      blank bottom=1,
      degree bottom=2
    ]{ % Upper non-transversal edges
      {1}{2}
    }{ % Lower non-transversal edges
      {2}{3}
    }{ % Transversal edges
      \pmdEmpty
    }
  }
}
```

The `options` allows for the redefinition of the command that draws the transversal edge. It has two arguments, the x-position of the edge in the top row of vertices and the x-position of the edge in the bottom row of vertices. The default is

```latex
\ifs
\pmdraw@ifTedgeHorizontal=0 % If drawing a straight line edge
  \draw[apply style/.expand once=\pmdraw@drawTedgesDrawOptions]
    (#1,\pmdraw@rowSep) -- (#2,0); % Draw straight line
\else % If transversal edge is drawn horizontally
  \ifs #1>\#2 % If edge goes from top right to bottom left
    \draw[apply style/.expand once=\pmdraw@drawTedgesDrawOptions]
      (#2,0) arc (180:90:\pmdraw{Tlevel}); % Draw bottom arch
    \draw[apply style/.expand once=\pmdraw@drawTedgesDrawOptions]
      (#2+\pmdraw{Tlevel},\pmdraw{Tlevel})
      -- (#1-\pmdraw@rowSep+\pmdraw{Tlevel}, \pmdraw{Tlevel}); % Draw straight line
  \else % If transversal edge is drawn horizontally
    \draw[apply style/.expand once=\pmdraw@drawTedgesDrawOptions]
      (#1-\pmdraw@rowSep+\pmdraw{Tlevel}, \pmdraw{Tlevel}) -- (#2,0) arc (180:90:\pmdraw{Tlevel}); % Draw bottom arch
    \draw[apply style/.expand once=\pmdraw@drawTedgesDrawOptions]
      (#1,\pmdraw@rowSep) -- (#2+\pmdraw{Tlevel},\pmdraw{Tlevel}); % Draw straight line
  \fi
\fi
```

21
\begin{pmdDiagram}
{ % Brick 1
  [ % Options
    degree top=5,
    degree bottom=6
  ]
  { % Upper non-transversal edges
    2\{3\}
    4\{5\}
  }
  { % Lower non-transversal edges
    3\{4\}
    5\{6\}
  }
  { % Transversal edges
    [options={
      \draw (#2,0) -- (#2+1,1); % Draw bottom line
      \draw (#2+1,1) -- (#1-1, 1); % Draw straight line
      \draw (#1-1,1) -- (#1,2); % Draw top line
    }]
    }\{1\}\{2\}
  \{1\}\{1\}
}
}{ % Brick 2
  [ % Options
    degree top=3,
    blank bottom=1,
    degree bottom=2
  ]
  { % Upper non-transversal edges
    1\{2\}
  }
  { % Lower non-transversal edges
    2\{3\}
  }
  { % Transversal edges
    \pmdEmpty
  }
}
\end{pmdDiagram}

An example use is

\begin{verbatim}
\pmdDiagram{
  { % Brick 1
    [ % Options
      degree top=5,
      degree bottom=6
    ]
    { % Upper non-transversal edges
      2\{3\}
      4\{5\}
    }
    { % Lower non-transversal edges
      3\{4\}
      5\{6\}
    }
    { % Transversal edges
      [options={
        \draw (#2,0) -- (#2+1,1); % Draw bottom line
        \draw (#2+1,1) -- (#1-1, 1); % Draw straight line
        \draw (#1-1,1) -- (#1,2); % Draw top line
      }]
      }\{1\}\{2\}
    \{1\}\{1\}
  }
}{ % Brick 2
  [ % Options
    degree top=3,
    blank bottom=1,
    degree bottom=2
  ]
  [ % Upper non-transversal edges
    1\{2\}
  ]
  [ % Lower non-transversal edges
    2\{3\}
  ]
  [ % Transversal edges
    \pmdEmpty
  ]
}
\end{verbatim}
3.5 Brace

**brick/brace** As part of a diagram, it may be useful to draw a brace. The options of a brace are as follows:

- **left** specifies the x-position of the start/left and end/right of the brace. Both must be provided.
- **right** specifies the label to be printed with the brace. Default is empty.

\pmdDiagram{
  \% Brick 1
  \% Options
  degree=2,
  brace={
    left=1,
    right=4,
    label={n}
  }
  \% Upper non-transversal edges
  \pmdEmpty
  \% Lower non-transversal edges
  \pmdEmpty
  \% Transversal edges
  \{1}\{1\}
  \{2}\{2\}
}

\pmdDiagram{
  \% Brick 2
  \% Options
  degree=1
  \% Upper non-transversal edges
  \pmdEmpty
  \% Lower non-transversal edges
  \pmdEmpty
  \% Transversal edges
  \{1}\{1\}
}

\above The **above** flag draws the brace above the diagram. Default is below.
options allows for the redefinition of the command that draws the brace. It has three arguments, the x-position start and end of the brace, as well as the label of the brace. The default is

\ifnum\pmdraw@ifProdDiag=1 % If drawing a product diagram
% or if placing brace above diagram
\draw[
  very thick,
  decorate,
  decoration={calligraphic brace,amplitude=6pt},
  apply style/.expand once=\pmdraw@drawBraceDrawOptions
] (#1-0.17,\pmdraw@rowSep+0.5) -- (#2+0.17,\pmdraw@rowSep+0.5)
node[
  pos=0.5,
  above=6pt,
  apply style/.expand once=\pmdraw@drawBraceNodeOptions
] {(#3\})}; % Draw brace on top of diagram
\else % If not drawing a product diagram or if placing brace below diagram
\fi
it should be noted that the flag \texttt{pmdraw@ifProdDiag} is both used for when drawing a product diagram but also when placing the brace above the diagram as this clearly must be the default for the top diagram in a product diagram. An example use is

\begin{verbatim}
\pmdDiagram{
  % Brick 1
  [ % Options
    degree=2,
    brace={
      left=1,
      right=4,
      label={n},
      options={
        \draw[
          red,
          very thick,
          decorate,
          decoration={calligraphic brace,mirror,amplitude=6pt},
          apply style/.expand once=\pmdraw@drawBraceDrawOptions
        ] (#1-0.17,-0.5) -- (#2+0.17,-0.5)
        node[ pos=0.5,
               below=6pt,
               apply style/.expand once=\pmdraw@drawBraceNodeOptions
        ] {\(#3\)}; % Draw brace on bottom of diagram
    }
  ]
}{ % Upper non-transversal edges
  \pmdEmpty
}{ % Lower non-transversal edges
  \pmdEmpty
}{ % Transversal edges
  \{1\}\{1\}
  \{2\}\{2\}
}{ % Brick 2
  [ % Options
    degree=1
  ]
}{ % Upper non-transversal edges
  \pmdEmpty
}{ % Lower non-transversal edges
  \pmdEmpty
}{ % Transversal edges
  \{1\}\{1\}
}
\end{verbatim}
3.6 Diagram options

3.6.1 For all (product) diagrams

The following options can be used for both diagrams (\pmdDiagram) and product diagrams (\pmdProduct).

- `row sep` same as row sep option in bricks but applied to all bricks within (product) diagram.
- `tikz` passes through options to tikz environment. Specifically, the options in \begin{tikzpicture}[<options>]\end{tikzpicture}. The default is no argument. However there are some default options passed to tikz automatically. These are an adjustment to the baseline depending on the current typesetting mode and a grid key if the grid is activated.

\begin{verbatim}
\pmdDiagram[ % Options
tikz={scale=2}
]{ % Brick 1
  [ % Options
    degree=2,
    brace={
      left=1,
      right=4,
      label={n}
    }
  ]
}{ % Upper non-transversal edges
  \pmdEmpty
}{ % Lower non-transversal edges
  \pmdEmpty
}{ % Transversal edges
  {1}{1}
  {2}{2}
}
}{ % Brick 2
  [ % Options
    degree=1
  ]
}{ % Upper non-transversal edges
  \pmdEmpty
}{ % Lower non-transversal edges
  \pmdEmpty
}{ % Transversal edges
  {1}{1}
}
}
\end{verbatim}
dots options allows for the redefinition of the command that draws dots between bricks. It has two arguments, the \textit{x} and \textit{y}-position of the dots. The default is

\begin{verbatim}
\draw[apply style/.expand once=\pmdraw@drawDotsDrawOptions]
(#1,#2) node[apply style/.expand once=\pmdraw@drawDotsNodeOptions] {\dots};
\end{verbatim}

An example use is

\begin{verbatim}
\pmdDiagram{ % Options
dots options={
  \draw[red] (#1,#2) node {\dots};
}
}%

\begin{verbatim}
{ % Brick 1
  [ % Options
    degree=2,
    brace={
      left=1,
      right=5,
      label={n}
    }
  ]% Upper non-transversal edges
  \pmdEmpty
  % Lower non-transversal edges
  \pmdEmpty
  % Transversal edges
  {1}{1}
  {2}{2}
}%

{ % Brick 2
  [ % Options
    degree=1,
    blank top=1,
    blank bottom=1
  ]% Upper non-transversal edges
  \pmdEmpty
  % Lower non-transversal edges
  \pmdEmpty
  % Transversal edges
  {2}{2}
}
}
\end{verbatim}
As it can get confusing where things are drawn and the different coordinate scopes between all the bricks and diagrams, the `grid` flag draws the base coordinate grid that the first drawn brick is based on. This should help with all coordinate related inquiries. Default is no grid.

It should be noted that the `grid` key is not available for \pmdBrick and \pmdDiagram must be used instead when needing a grid for a single brick only.

\pmdDiagram[ % Options
grid
]

\{ % Brick 1
  [ % Options
degree top=5,
degree bottom=6,
  ]\{ % Upper non-transversal edges
  \{2\}{3}
  \{4\}{5}
\}
\{ % Lower non-transversal edges
  \{3\}{4}
  \{5\}{6}
\}
\{ % Transversal edges
  [height=1]\{10\}{2}
  \{1\}{1}
\}
\}

\{ % Brick 2
  [ % Options
degree top=3,
  blank bottom=1,
  degree bottom=
  ]\{ % Upper non-transversal edges
  \{1\}{2}
\}
\{ % Lower non-transversal edges
  \{2\}{3}
\}
\{ % Transversal edges
  \pmdEmpty
\}
\}

\decorate before same as \decorate before and \decorate after in bricks but these are drawn before (after) all bricks are drawn. For clarity, this is default drawing order for a diagram:

1. diagram/decorate before
2. Brick loop:
   (a) brick/decorate before
   (b) Draws brick
   (c) brick/decorate after

And this is default drawing order for a product diagram:

1. diagram/decorate before
2. Draws added edges
3. Top diagram:
   (a) Brick loop:
      i. brick/decorate before
      ii. Draws brick
      iii. brick/decorate after
4. Bottom diagram:
   (a) Brick loop:
      i. brick/decorate before
      ii. Draws brick
      iii. brick/decorate after

5. diagram/decorate after

For the drawing order of a brick see Section 3.3.5.

3.6.2 For product diagrams only

The following options can only be used for product diagrams (\pmdProduct).

\texttt{diagram sep} specifies the vertical separation between the bottom row of vertices in top diagram and the top row of vertices in bottom diagram. Default is one.
\pmdProduct{ % Options
diagram sep=2
}{ % Added edges
(1)(5)
}{ % Top diagram
{ % Brick 1
  [ % Options
dergee=5
  ]{ % Upper non-transversal edges
  \pmdEmpty
  }{ % Lower non-transversal edges
  \pmdEmpty
  }{ % Transversal edges
    (1)(1)
    (2)(3)
    (3)(2)
    (4)(4)
    (5)(5)
  }
}
}{ % Bottom diagram
{ % Brick 1
  [ % Options
dergee=5
  ]{ % Upper non-transversal edges
    (2)(3)
    (4)(5)
  }{ % Lower non-transversal edges
    (2)(3)
    (4)(5)
  }{ % Transversal edges
    (1)(1)
  }
}
}

\begin{tikzpicture}
  \draw (1,0) -- (5,0);
  \draw (1,1) -- (5,1);
  \draw (1,2) -- (5,2);
  \draw (1,3) -- (5,3);
  \draw (1,4) -- (5,4);
  \draw (1,5) -- (5,5);
\end{tikzpicture}

\textbf{edges added} passes through options to the \texttt{draw} command of the added edges. The default is no argument.
4 Future features/work

I have attempted to make most use cases easy to use with dedicated functionality that does not require significant \LaTeX{} knowledge whilst at the same time provide more advanced interfaces to allow for hopefully almost complete customisability for those who put in the effort.

Having said that, please do let me know if there are features or improvements you will like to see added.

The following are some of the aims for the future for this package:
1. Fix bugs.

2. Add warning and error messages to log.

3. Allow all brick keys to work as diagram keys that apply to all bricks in a diagram. This will require setting up a flag for each key so that the diagram key sets the brick key but the brick key is not reset after drawing the first brick. This process also will need to be able to handle when a single brick has a custom special different key value and that this is reset to the diagram key value and not the global default key value.

4. Allow appropriate diagram keys to work as brick keys. Notably the \texttt{grid} key.

5. The creation of user profiles that allows for fast and efficient setting of default key values that are different to the package defaults. Hence within a diagram, there can be a number of different diagram styles that can be accessed with one command.

6. Allow for adjusting the horizontal spacing of vertices. Need to think about making that a simple scale factor so that integer interface for edge positions is the same but then hooks will have very messy coordinates or keep it as absolute coordinates with messy edges. Alternatively, do both, allowing the user to choose which option to do and somehow distinguish between the two in the background.

5 Examples

This section is simply a collection of example diagrams. The code for them can be found in the \texttt{examples} folder. Perhaps you will find them useful as a starting point for your diagrams or to see what this package is capable of.

Example 1

\begin{center}
\includegraphics[width=0.2\textwidth]{example1.png}
\end{center}

Example 2

\begin{center}
\includegraphics[width=0.2\textwidth]{example2.png}
\end{center}

Example 3

\begin{center}
\includegraphics[width=0.2\textwidth]{example3.png}
\end{center}
Example 4

Example 5

Example 6

Example 7

Example 8

Example 9