Abstract
The 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 ............................................... 16
    3.3.3 Brace ......................................................... 19
    3.3.4 Decorations .................................................. 21
    3.3.5 Drawing order ............................................... 23
  3.4 Edge options ........................................................ 23
    3.4.1 Non-transversal edge options ............................... 23
    3.4.2 Transversal edge options ................................... 26
  3.5 Brace ............................................................... 29
  3.6 Diagram options .................................................... 33
    3.6.1 For all (product) diagrams .................................. 33
    3.6.2 For product diagrams only .................................. 36
4 Future features/work 41
5 Examples 43

*This document corresponds to pmdraw v1.4, 2024-06-02.
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/642555).

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 level or height that makes the arcs smaller.

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

3. In a product diagram, if row sep is used in the bottom diagram, the spacing between diagrams is incorrect.

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

brick A brick is a diagram that contains no dots. For example,

\[
\begin{array}{c}
\bullet & \bullet & \bullet \\
\end{array}
\]

is a brick whilst

\[
\begin{array}{c}
\bigcirc & \bigcirc & \cdots & \bigcirc \\
\end{array}
\]

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

<brick> = [{options}][{Uedges}]{Ledges}{Tedges}

brick/<options> where the <options> are described in Section 3.3.

<Uedges> is a list of upper non-transversal edges. An upper non-transversal edge has three components <Uedges> = [{options}]{Svertex}{Tvertex} where the <options> are described in Section 3.4.1, {Svertex} is the starting vertex and {Tvertex} is the terminating vertex of the edge. Both {Svertex} and {Tvertex} are a -coordinate and it is assumed that {Svertex} < {Tvertex}.

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

\texttt{\textbackslash pmdEmpty} Should a list be empty, the command \texttt{\textbackslash pmdEmpty} should be used.
\texttt{\textbackslash Ledges} is a list of lower non-transversal edges. A lower non-transversal edges has the same three components as an upper non-transversal edge, namely
\\[\texttt{\textbackslash Ledges} = [\texttt{\langle options\rangle}]{\langle Svertex\rangle}{\langle Tvertex\rangle}].\]
\texttt{\textbackslash Tedges} is a list of transversal edges. A transversal edges has three components
\\[\texttt{\textbackslash Tedges} = [\texttt{\langle options\rangle}]{\langle Tvertex\rangle}{\langle Bvertex\rangle}].\]

An example \texttt{\langle brick\rangle} and the resultant diagram is:
\begin{verbatim}
\texttt{\textbackslash pmdBrick[ % Options}
\texttt{ degree=5}
\texttt{ ]}{ % Upper non-transversal edges}
\texttt{ {1\{2\}}\{2\}5\}}{ % Lower non-transversal edges}
\texttt{ {1\{2\}}\{3\}4\}}{ % Transversal edges}
\texttt{ \textbackslash pmdEmpty}
\texttt{ }
\end{verbatim}

3.2 Draw commands

This package has three draw commands.
\texttt{\textbackslash pmdBrick} draws a brick and has the syntax \texttt{\textbackslash pmdBrick\langle brick\rangle}.
\begin{verbatim}
\texttt{\textbackslash pmdBrick[ % Options}
\texttt{ degree=5}
\texttt{ ]}{ % Upper non-transversal edges}
\texttt{ {1\{2\}}\{2\}5\}}{ % Lower non-transversal edges}
\texttt{ {1\{2\}}\{3\}4\}}{ % Transversal edges}
\texttt{ \textbackslash pmdEmpty}
\texttt{ }
\end{verbatim}
\texttt{\textbackslash pmdDiagram} draws arbitrarily many bricks separated by dots and has the syntax
\begin{verbatim}
\texttt{\textbackslash pmdDiagram[\langle options\rangle]\{\langle bricks\rangle\}}
\texttt{ }
\end{verbatim}
diagram/\langle options\rangle where the \langle options\rangle are described in Section 3.6 and \langle bricks\rangle is a list of \langle bricks\rangle \langle brick\rangle-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
    degree=3,
    brace={
      left=2,
      right=6,
      label={n}
    }
  ]% Upper non-transversal edges
  \pmdEmpty
  % Lower non-transversal edges
  \pmdEmpty
  % Transversal edges
  {1}{1}
  {2}{6}
  {3}{5}
}

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

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

  \pmdProduct
  diagram/<options> 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>}.

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

  <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.
It should also be noted that the \texttt{blank top} and \texttt{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 $-1$ and the dots is redefined to not draw, the space between bricks can be removed altogether. See Examples 13-15.

\texttt{row sep} specifies the vertical separation between rows of vertices. Default is two. If using edge options such as \texttt{levels}, \texttt{row sep} must be given before \texttt{levels}.

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

\texttt{no vertex bottom 1} specifies a vertex that is not drawn. Any of the nine available keys can be used in any order. It can be used to draw single vertices that are different to the others. If more than nine vertices need to be excluded, simply draw multiple bricks next to each other. The default is no argument.
\pmdBrick[ % Options
degree=5,
no vertex top 1=2,
no vertex top 2=3,
no vertex bottom 1=4,
decorate after=
\fill[red] (4,0) circle (.1);
]{ % 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 \textit{x} and \textit{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}
}

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

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

labels top start
labels bottom start \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\}
}

no labels bottom 1 specifies a vertex for which no label is drawn. Any of the nine available keys can be used in any order. It can be used to insert single unique labels that are different to the others. If more than nine labels need to be excluded, simply draw multiple bricks next to each other. The default is no argument.

no labels bottom 9
no labels top 1
no labels top 9 \pmdBrick[ % Options
degree=5,
labels,
no label top 1=2,
no label top 2=3,
no label bottom 1=4,
decorate after={
  \draw (3,2) node[above=6pt,red] \{\(x\)\};
}
]{ % Upper non-transversal edges
  \{1\}(2)
  \{2\}(5)
\}{ % Lower non-transversal edges
  \{1\}(2)
  \{3\}(4)
\}{ % Transversal edges
  \{1\}\{1\}
  \{3\}\{4\}
}
labels top draw \draw and node command of the labels of vertices. The default is no argument.
labels bottom draw
labels bottom node

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
\draw[apply style/.expand once=\pmdraw@drawLabelTDrawOptions] (#1,#2)
node[above=6pt,apply style/.expand once=\pmdraw@drawLabelTNodeOptions] {(#3)};

and for the bottom
\draw[apply style/.expand once=\pmdraw@drawLabelBDrawOptions] (#1,#2)
node[below=6pt,apply style/.expand once=\pmdraw@drawLabelBNodeOptions] {(#3')};

An example use is
\begin{document}

\pmdBrick[ % Options
degree=5,
labels,
labels bottom options={
  \draw (#1,#2) node[below=6pt,blue] {(#3''\});
}
]{ % Upper non-transversal edges
  (1){2}
  (2){5}
}{ % Lower non-transversal edges
  (1){2}
  (3){4}
}{ % Transversal edges
  (1){1}
  (3){4}
}

\begin{tikzpicture}
\node (A) at (0,0) {1};
\node (B) at (1,0) {2};
\node (C) at (2,0) {3};
\node (D) at (3,0) {4};
\node (E) at (4,0) {5};
\node (A') at (0,-1) {$1''$};
\node (B') at (1,-1) {$2''$};
\node (C') at (2,-1) {$3''$};
\node (D') at (3,-1) {$4''$};
\node (E') at (4,-1) {$5''$};
\draw (A) -- (B);
\draw (B) -- (C);
\draw (C) -- (D);
\draw (D) -- (E);
\draw (A') -- (B');
\draw (B') -- (C');
\draw (C') -- (D');
\draw (D') -- (E');
\end{tikzpicture}

\noindent \textbf{no dots} \quad The \textbf{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.

\end{document}
dots draw passes through options to the \draw and node command of the dots. The dots node default is no argument.
These flags allow for all of the top or bottom vertices to be hidden using the \texttt{vertices top phantom} and \texttt{vertices bottom phantom} command. Simply wraps the vertex drawing macros with a \texttt{phantom} command. This hiding will also apply to the labels of that row (top or bottom) of vertices. To only hide a selected number of vertices, use separate bricks. The default is false.

\begin{verbatim}
\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,
    dots node=red
  ]% Upper non-transversal edges
  \pmdEmpty
  ]% Lower non-transversal edges
  \pmdEmpty
  ]% Transversal edges
  {1}{1}
}
}
\end{verbatim}
\textbf{vertices top uncover} allows for all of the top or bottom vertices to be uncovered in a \texttt{beamer} presentation using the command $\texttt{uncover<n->{text}}$. Simply passes the slide uncover information $\langle n \rangle$ to the embedded $\texttt{uncover}$ command within the vertex drawing macros. This uncover will also apply to the labels of that row (top or bottom) of vertices. To only uncover a selected number of vertices, use separate bricks. The default is 1-.

\begin{verbatim}
\pmdBrick[ % Options
degree=5,
vertices top uncover={2,4-7,9-},
vertices bottom uncover={3-}
]{ % Upper non-transversal edges
  (1){2}
  (2){5}
}{ % Lower non-transversal edges
  (1){2}
  (3){4}
}{ % Transversal edges
  (1){1}
  (3){4}
}
\end{verbatim}

\subsection*{3.3.2 Edge options}

These options should be read in conjunction with Section 3.4.

\texttt{levels} specifies how many horizontal levels there are for non-transversal edges. Default is one.

\begin{verbatim}
\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
}\end{verbatim}

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

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.

\pmdBrick[ % Options
degree=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
}

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.

\pmdBrick[ % Options
degree=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
}

edges pass through options to the \draw command of their respective edge types.
edges non-transversal The default is no argument.
edges upper
edges lower
edges transversal
\pmdBrick[ % Options 
\begin{itemize}
  \item degree=5,
  \item edges upper=blue,
  \item edges lower={red, dotted, very thick},
  \item edges transversal=dashed
\end{itemize}
]{ % Upper non-transversal edges
\begin{itemize}
  \item (1){2}
  \item (2){5}
\end{itemize}
}{ % Lower non-transversal edges
\begin{itemize}
  \item (1){2}
  \item (3){4}
\end{itemize}
}{ % Transversal edges
\begin{itemize}
  \item (1){1}
  \item (3){4}
\end{itemize}
}

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

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

transversals first flag ensures transversal edges are drawn before non-transversal edges. The default is non-transversal edges are drawn first.

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

edges phantom This flag allows for all edges to be hidden using the \texttt{phantom} command. Simply wraps the edge drawing macros with a \texttt{phantom} command. The default is false.
\pmdBrick[ % Options
degree=5,
edges phantom
]{ % Upper non-transversal edges
  {1}{2}
  {2}{5}
}{ % Lower non-transversal edges
  {1}{2}
  {3}{4}
}{ % Transversal edges
  {1}{1}
  {3}{4}
}

edges uncover allows for all edges to be uncovered in a \texttt{beamer} presentation using the command \texttt{uncover<\textit{n->text}}. Simply passes the slide uncover information <n-> to the embedded \texttt{uncover} command within the edge drawing macros. The default is 1-.

\pmdBrick[ % Options
degree=5,
edges uncover={2,4-7,9-}
]{ % 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

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.
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.
3.3.4 Decorations

decorate before provides a hook to draw additional \texttt{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 \texttt{tikz} elements but arbitrary code before and after draw a brick.

\begin{pndDiagram}
\begin{pndTopRow}
\begin{pndBrick}[ % Options
    degree=4,
    decorate after={
        \draw[
            very thick, red, rounded corners=5pt
        ] (0.6, -0.4) rectangle (1.4, 2.4);
    }
\end{pndBrick}
\begin{pndBrick}[ % Options
    degree=3,
    decorate before={\begin{scope}[red,shift={(1,1)},rotate=30]},
    decorate after={\end{scope}}
\end{pndBrick}
\end{pndTopRow}
\end{pndDiagram}
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.

```latex
\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.
\pmdBrick[ % Options
degree=5
]{ % Upper non-transversal edges
  (1)(5)
  [height=1.75](1)(4)
}{ % Lower non-transversal edges
  [height=1.25](1)(5)
  (1)(4)
}{ % Transversal edges
\pmdEmpty
}

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

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

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

and for a lower non-transversal

\draw[apply style/.expand once=\pmdraw@drawLedgesDrawOptions, apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
  (#1,0) arc (180:90:\pmdraw{edgeHeight});% Draw left arch
\draw[apply style/.expand once=\pmdraw@drawLedgesDrawOptions, apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
  (#1+\pmdraw{edgeHeight},\pmdraw{edgeHeight})
  -- (#2-\pmdraw{edgeHeight},\pmdraw{edgeHeight});% Draw straight line
\draw[apply style/.expand once=\pmdraw@drawLedgesDrawOptions, apply style/.expand once=\pmdraw@drawEdgeDrawOptions]
An example use is
\pmdBrick[ % Options
degree=5
]{ % Upper non-transversal edges
 {1}{2}
 {2}{5}
}{ % Lower non-transversal edges
 [options={
 \draw[blue, very thick] (#1,0) -- (#2,0);
 }]{1}{2}
 {3}{4}
}{ % Transversal edges
 {1}{1}
 {3}{4}

This flag allows for the edge to be hidden using the \texttt{phantom} command. Simply wraps the edge drawing macros with a \texttt{phantom} command. The default is false.

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

allows for that edge to be uncovered in a \texttt{beamer} presentation using the command \texttt{\uncover<n->\{text\}}. Simply passes the slide uncover information \texttt{<n->} to the embedded \texttt{uncover} command within the edge drawing macros. This uncover is nested inside the \texttt{edges uncover} and hence the limitations of nested uncovers are present here. For example, an individual edge will not uncover until the \texttt{edges uncover} has been uncovered. In such instances only use individual uncovers for each edge. The default is \texttt{1-}.
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.

\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
    [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
  }
}
\end{verbatim}

The `edge draw` option passes through options to the `\draw` command of the edge. The default is no argument.
\pmdBrick[ % Options
degree=5
]{ % Upper non-transversal edges
{1}{2}
}{ % Lower non-transversal edges
{4}{5}
{3}{4}
}{ % Transversal edges
\begin{tikzpicture}
\draw[red,thick] (1,-2) -- (5,-2);
\draw[red,thick] (1,-2) -- (5,-2);
\draw[red,thick] (1,-2) -- (5,-2);
\end{tikzpicture}
}

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

\begin{verbatim}
\ifnum\pmdraw@ifTedgeHorizontal=0% If drawing a straight line edge
\draw[\pmdraw@drawTedgesDrawOptions, \pmdraw@drawEdgeDrawOptions] (#1,\pmdraw@rowSep) -- (#2,0);% Draw straight line
\else% If transversal edge is drawn horizontally
\ifnum#1>#2% If edge goes from top right to bottom left
\draw[\pmdraw@drawTedgesDrawOptions, \pmdraw@drawEdgeDrawOptions] (#2,0) arc (180:90:\pmdraw{Tlevel});% Draw bottom arch
\draw[\pmdraw@drawTedgesDrawOptions, \pmdraw@drawEdgeDrawOptions] (#2+\pmdraw{Tlevel},\pmdraw{Tlevel}) -- (#1-\pmdraw@rowSep+\pmdraw{Tlevel}, \pmdraw{Tlevel});% Draw straight line
\else% If edge goes from top left to bottom right
\draw[\pmdraw@drawTedgesDrawOptions, \pmdraw@drawEdgeDrawOptions] (#2,0) arc (0:90:\pmdraw{Tlevel});% Draw bottom arch
\draw[\pmdraw@drawTedgesDrawOptions, \pmdraw@drawEdgeDrawOptions] (#2-\pmdraw{Tlevel},\pmdraw{Tlevel}) -- (#1+\pmdraw@rowSep-\pmdraw{Tlevel}, \pmdraw{Tlevel});% Draw straight line
\fi
\fi
\renewcommand{\pmdraw@ifTedgeHorizontal}{0}% Reset flag to default
\fi
\renewcommand{\pmdraw@drawEdgeDrawOptions}{}% Reset draw options for edge
\end{verbatim}

An example use is

27
This flag allows for the edge to be hidden using the \texttt{phantom} command. Simply wraps the edge drawing macros with a \texttt{phantom} command. The default is false.

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

\texttt{uncover} allows for that edge to be uncovered in a \texttt{beamer} presentation using the com-
mand `\uncover<n->{text}`. Simply passes the slide uncover information `<n->` to the embedded `uncover` command within the edge drawing macros. This uncover is nested inside the `edges uncover` and hence the limitations of nested uncover are present here. For example, an individual edge will not uncover until the `edges uncover` has been uncovered. In such instances only use individual uncovers for each edge. The default is `1-`.

\pmdBrick[ % Options
degree=5
]{ % Upper non-transversal edges
  (1){2}
  (2){5}
}{ % Lower non-transversal edges
  (1){2}
  (3){4}
}{ % Transversal edges
  {uncover={2,4-7,9-}}{1}{1}
  (3){4}
}

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

```
\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)
        };
    }
  }
  % 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}

}
```
3.6 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}\[<options>\]. 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}

33
dots options allows for the redefinition of the command that draws dots between bricks. It has two arguments, the x and y-position of the dots. The default is

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

An example use is

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

3. **diagram/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**).

`\text{diagram sep}` specifies the vertical separation between the bottom row of vertices in top diagram and the top row of vertices in bottom diagram. Also controls the length of the added edges. Default is one.
\begin{verbatim}
\pmdProduct[ % Options
diagram sep=2]
}{ % Added edges
  \{1\}{5}\}
}{ % 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}
  \}
  \}
}{ % 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}
  \}
  \}
\end{verbatim}

\texttt{vertex sep} increases the vertical separation between the bottom row of vertices in top diagram and the top row of vertices in bottom diagram without altering the length of the added edges. Useful for things when adjusting the \texttt{row sep} of the bottom diagram for example. Default is zero.
\pmdProduct{ % Options
  \vertexSep=1
}{ % Added edges
  \{1\}(5)
}{ % 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\}
    \}
  \}
}{ % Bottom diagram
  \{ % Brick 1
    \{ % Options
      degree=5,\rowSep=3
    \}{ % Upper non-transversal edges
      \{2\}\{3\}
      \{4\}\{5\}
    }\{ % Lower non-transversal edges
      \{2\}\{3\}
      \{4\}\{5\}
    }\{ % Transversal edges
      \{1\}\{1\}
    \}
  \}
}

\begin{center}
\begin{tikzpicture}
  \draw (-1,0) -- (1,0) node[fill,draw] {\ldots} -- (1,2) node[fill,draw] {\ldots} -- (-1,2) node[fill,draw] {\ldots} -- cycle;
  \draw (-1,1) -- (1,1) node[fill,draw] {\ldots} -- (1,3) node[fill,draw] {\ldots} -- (-1,3) node[fill,draw] {\ldots} -- cycle;
\end{tikzpicture}
\end{center}

*edges added* passes through options to the `\draw` command of the added edges. The default is no argument.
This flag allows for the added edges to be hidden using the \texttt{phantom} command. Simply wraps the edge drawing macros with a \texttt{phantom} command. The default is false.
\pmdProduct{ % Options
edges added phantom
}{ % Added edges
  (1){5}
}{ % 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}
    }
  }
}{ % Bottom diagram
  % Brick 1
    [ % Options
degree=5
    ]{ % Upper non-transversal edges
    }{ % Lower non-transversal edges
    }{ % Transversal edges
      {1}{1}
    }
  }

allows for the added edges to be uncovered in a \texttt{beamer} presentation using the command \texttt{\uncover<n->{text}}. Simply passes the slide uncover information \texttt{<n->} to the embedded \texttt{\uncover} command within the added edge drawing macros. The default is \texttt{1-}. 

40
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 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. That is, in the options for a diagram, have this profile key.

6. Have global settings so that all diagrams in a document have the specified options.

7. 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.

8. Create `pmdMirror` command that mirrors all upper non-transversals into lower non-transversals.

9. Create `invert` key for bricks, diagrams and product diagrams that inverts everything. Probably easiest done with a `scope` environment around everything and `yscale=-1`.

10. For product diagram, have interface where you input diagram instead of bricks so that you have access to diagram options.

11. Create a transversal edge key `identity` so that all edges in between the two values are straight vertical edges. Thus eliminating the need to write out each edge manually.

12. Create `shift` key for edges that shifts horizontal position.

13. Decouple the uncover of labels from `vertices top uncover/vertices bottom uncover`.

14. Decouple the phantom of labels from `vertices top phantom/vertices bottom phantom`.
5 Examples

This section is simply a collection of example diagrams. The code for them can be found in the 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

Example 2

Example 3

Example 4

Example 5

Example 6
Example 19

Example 20

Example 21

Example 22

46
Example 23

Example 24

Example 25

$p - 3$