

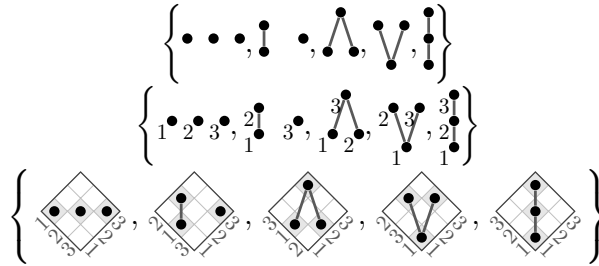
package

causets

version 1.3

Christoph Minz

May 18, 2022



What is the purpose of this package?

Causal set theory is a proposed framework for quantum gravity. Causal sets (**causets** for short) are locally finite, partially ordered sets. Though one can describe them mathematically as a set with a partial order, it is usually much simpler to just draw (Hasse) diagrams.

This package implements some simple functionality to produce such diagrams for causets by drawing them as TikZ graphics. The drawing algorithm makes use of the fact that every finite causet that embeds in $1 + 1$ dimensional Minkowski spacetime can be represented by a product of two total orders (2D-orders), where one order can be the total order of the integers labelling the causet elements, while the other is a permutation of the labels¹. The idea is then used as a blueprint to draw diagrams for any generic (finite) causet, so that it becomes easy to draw causets and even use them in mathematical expressions.

¹See <https://arxiv.org/abs/0706.0375> and <https://arxiv.org/abs/2011.02965> for more details on 2D-orders and causal sets.

1 Introduction - the main commands

This package is built on the package `tikz` to draw the diagrams. To import the `causets` package into a \LaTeX file, write

```
1 \usepackage{causets}
```

in the preamble of the document (this will automatically load the `tikz` package as well).

For the command introduction in [Table 1](#), I use the package options `black` (only black and white diagrams) and `permutation` (also shows the permutation),

```
1 \usepackage[black,permutation]{causets}
```

At first, let us take a brief look at the three main commands of the package, before we look into the (physical) ideas behind these three commands.

The three main commands of this package are

```
1 \pcauset{... ,i,... ,k,... ,j,... }
2 \rcauset{... }{... ,i/j,... }
3 \causet{... }{... ,i/j,... }
```

The first command takes a permutation of the integers from an interval $[a, b]$ (where $N = b - a + 1 \geq 0$ is the cardinality of the causet). The permutation determines the event positions and causal structure such that events i and j are linked if $i < j$ and there is no $k > j$ between i and j in the permutation. The second command does the same, but ignores links between event pairs i/j as given in the second argument. The third command only uses the permutation to position the events, the links have to be explicitly specified by event pairs i/j in the second argument.

Any causet that can be embedded in $1 + 1$ dimensional Minkowski spacetime can also be described by the product of two total orders (a 2D-order). Let the causets elements be labeled by an increasing null coordinate u . In the first diagram of [Table 1](#), the u -coordinate axis points upwards on the right hand side. The causal relation (partial order) implies a label permutation along the null coordinate v (pointing upwards along the left axis). This permutation is the input of the `\pcauset` command that computes the causal relation from the permutation. See example 1 in [Table 1](#).

Every causet that does not embed in $1 + 1$ dimensional Minkowski spacetime can be made embeddable by adding links, for which the `\rcauset` command is defined. Find an embedding causet by adding a minimal number of causal relations to the given causet. The so obtained, embedding causet gives a permutation as discussed before for the `\pcauset` causet. The links that have been added to obtain this permutation need to be removed from the generated causet, which is specified by link pairs i/j in the second argument. See example 2 in [Table 1](#).

In some situations, it might be easier to specify the list of link pairs that exist, instead of removing links. For this purpose, use the `\causet` command. As an example, we rewrite the example 2 by specifying all existing links as pairs, see example 3 in [Table 1](#). Note that the `\rcauset` command was actually much shorter for this particular example. However,

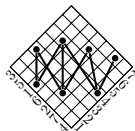

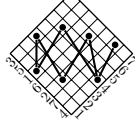
#	command	inline output
1	<code>\pcauset{4,7,2,6,1,5,3}</code>	
2	<code>\rcauset{4,7,2,6,1,5,3}{2/5}</code>	
3	<code>\causet{4,7,2,6,1,5,3}{1/3,1/5,2/3,2/6,4/5,4/6,4/7}</code>	

Table 1: Examples for the three main commands of the package.

if the list of link pairs in the `\rcauset` command is very long, specifying all existing link pairs instead with the `\causet` yields a faster performance.

You may use any of the three main commands to add a causet diagram in-line with text or within a mathematical expression. As an example for the use in a mathematical equation, here is the code for the first set of causets on the title page:

```

1 \usepackage{causets}
2
3 ...
4
5 \begin{document}
6
7 ...
8
9 \begin{align}
10 S &=
11 \left\{
12 \pcauset{3,2,1}, \pcauset{3,1,2}, \pcauset{2,1,3}, \pcauset{1,3,2},
13 \pcauset{1,2,3}
14 \right\}.
\end{align}

```

$$S = \{\cdots, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot\}. \quad (1)$$

Of course, a given causet might have more than one representing permutation, so that any of those permutations may be used. However, you may consider it a good practice to choose the permutation that starts with the larger integer, so that disjoint elements appear to the right in the graphs. For example, you may write `\pcauset{3,1,2}, \cdot, \cdot`, as opposed to `\pcauset{2,3,1}, \cdot, \cdot`.

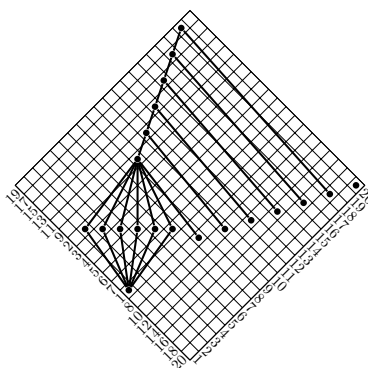
The arguments of the commands are lists that are iterated over by a `\foreach` loop of PGF/TikZ. So you may use dots `...` to complete ranges of integers automatically. You

may use this feature to draw chains like \vdots (with `\pcauset{1,\dots,5}`) or antichains like \dots (with `\pcauset{5,\dots,1}`). As a more complex example for the dots notation, consider

```
1 \pcauset{20,18,\dots,8,
2     1,7,6,\dots,2,
3     9,11,\dots,19}
```


which produces the same diagram as

```
1 \pcauset{20,18,16,14,12,10,8,
2     1,7,6,5,4,3,2,
3     9,11,13,15,17,19}
```



2 Package options

Each of the package options can either be specified in square brackets when declaring the package (option applies globally) or be changed with a macro at any point in the document (option applies to the local scope).

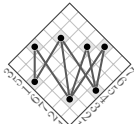
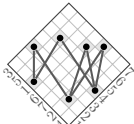
If no options are used, an output like  is obtained with the commands

```
1 \usepackage{causets}
2 ... \begin{document} ...
3 \rcauset{4,2,7,6,1,5,3}{2/5}
```

The options – given in the following table – are passed to the `causets` package as optional parameter and can be combined. The package options are listed together with example outputs created with the commands

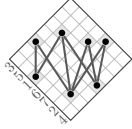
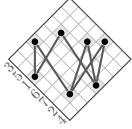
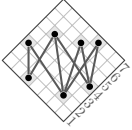















```
1 \pcauset{4,2,7,6,1,5,3}
2 \rcauset{4,2,7,6,1,5,3}{2/5}
```

If not specified otherwise, any package option also works with the command `\causet`.

Package option	<code>\pcauset</code>	<code>\rcauset</code>
<code>\usepackage[permutation]{causets}</code> shows permutations (<code>\causetsDrawPermutationtrue</code> activates this option) (<code>\causetsDrawPermutationfalse</code> deactivates this option)		


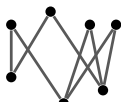
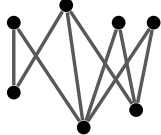
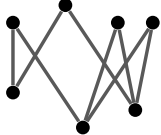










continued on the next page

continued from the previous page

Package option	<code>\pcauset</code>	<code>\rcauset</code>
<code>\usepackage[permutation, unlabelled u]{causets}</code> or <code>\usepackage[permutation, unlabeled u]{causets}</code> shows permutations without u-labels (<code>\causetsDrawULabelstrue</code> activates u-labels) (<code>\causetsDrawULabelsfalse</code> deactivates u-labels)		
<code>\usepackage[permutation, unlabelled v]{causets}</code> or <code>\usepackage[permutation, unlabeled v]{causets}</code> shows permutations without v-labels (<code>\causetsDrawVLabelstrue</code> activates v-labels) (<code>\causetsDrawVLabelsfalse</code> deactivates v-labels)		
<code>\usepackage[labelled]{causets}</code> or <code>\usepackage[labeled]{causets}</code> shows labeled events (<code>\causetsDrawLabelstrue</code> activates labels) (<code>\causetsDrawLabelsfalse</code> deactivates labels)		
<code>\usepackage[continuously linked]{causets}</code> allows links to cross over events. Broken links (that is the default option: <code>\usepackage[brokenly linked]{causets}</code>) affects only the <code>\causet</code> command, but not the examples of <code>\pcauset</code> and <code>\rcauset</code> , shown on the right. Details on broken links are given below.		
<code>\usepackage[cap linked]{causets}</code> sets caps as link tips		
<code>\usepackage[arrow linked]{causets}</code> sets arrows (rounded, open stealth) as link tips		
<code>\usepackage[unlinked]{causets}</code> does not draw links (<code>\causetsDrawLinkstrue</code> activates links) (<code>\causetsDrawLinksfalse</code> deactivates links)		
<code>\usepackage[tiny]{causets}</code> for tiny diagrams (<code>\tikzcausetsset{tiny}</code> sets tiny sizes)		
<code>\usepackage[small]{causets}</code> for small diagrams (<code>\tikzcausetsset{small}</code> sets small sizes)		

continued on the next page

continued from the previous page

Package option	<code>\pcauset</code>	<code>\rcauset</code>
<code>\usepackage[large]{causets}</code> for large diagrams (<code>\tikzcausetsset{large}</code> sets large sizes)		
<code>\usepackage[huge]{causets}</code> for huge diagrams (<code>\tikzcausetsset{huge}</code> sets huge sizes) (<code>\tikzcausetsset{normal}</code> resets to normal sizes) Details on size options are given in Sec. 5 .		
<code>\usepackage[black]{causets}</code> for black and white causets (<code>\tikzcausetsset{black colours}</code> activates the option)		
<code>\usepackage[gray]{causets}</code> or <code>\usepackage[gray]{causets}</code> for grey scale causets (default) (<code>\tikzcausetsset{grey colours}</code> or <code>\tikzcausetsset{gray colors}</code> activates the option)		
<code>\usepackage[blue]{causets}</code> for blue causets (<code>\tikzcausetsset{blue colours}</code> activates the option)		
<code>\usepackage[neon]{causets}</code> for brightly coloured causets (<code>\tikzcausetsset{neon colours}</code> activates the option) Details on colour options are given in Sec. 6 .		
<code>\usepackage[spatially linked]{causets}</code> shows dashed lines between spatially separated events (which is a secondary partial order for the causets drawn with <code>\pcauset</code>). This option only affects the <code>\pcauset</code> and <code>\rcauset</code> commands, but not the <code>\causet</code> command. (<code>\causetsDrawSpatialLinkstrue</code> activates the option) (<code>\causetsDrawSpatialLinksfalse</code> deactivates the option)		

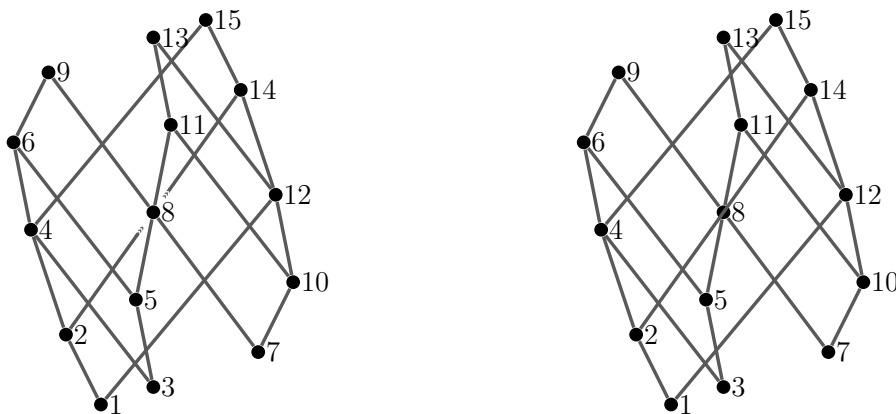
By default, the links in a `\causet` diagram are interrupted if they cross over another event (exactly over the event center). Any such crossing of links over events can only be caused by the manually drawn links, explicitly given in the second argument of `\causet`, but not with the links drawn by `\pcauset` or `\rcauset`. This feature is activated by default (`\causetsBreakLinkstrue`) and can be deactivated with `\causetsBreakLinksfalse`, or with the option `continuously linked`.

For example, here is a causet that embeds in 3-dimensional but not in 2-dimensional Minkowski spacetime. This causet has one event (0) in the center that is linked to events -3 and 3 but not to events -6 and 6, while a link from -6 to 6 crosses over it.

```

1 \usepackage[huge]{causets}
2 \tikzcausetsset{labels={right, position=east}, offset=-8}
3
4 ... \begin{document} ...
5
6 \causet[brokenly linked]{7,10,3,1,12,5,2,8,14,11,4,15,13,6,9}
   {1/2,1/12,2/4,2/14,3/4,3/5,4/6,4/15,5/6,5/8,6/9,7/8,7/10,8/9,8/11,
   10/11,10/12,11/13,12/13,12/14,14/15}
7 \causet[continuously linked]{7,10,3,1,12,5,2,8,14,11,4,15,13,6,9}
   {1/2,1/12,2/4,2/14,3/4,3/5,4/6,4/15,5/6,5/8,6/9,7/8,7/10,8/9,8/11,
   10/11,10/12,11/13,12/13,12/14,14/15}

```



Note that the option `brokenly linked` is not necessary as long as the default option has not been changed.

3 Short hand macros





The permutations and labels for causets can be switched on globally with the package options `permutation` and `labeled`, or at any point in the document with the switches `\causetsDrawPermutationtrue` (same as the command `\tikzcausetsset{show permutation}` used in the preamble) and `\causetsDrawLabelstrue` (or `\tikzcausetsset{labeled}` used in the preamble), respectively.

To show only single causets with either of these options, there exist short hand macros for all three main commands. Each main command can be followed by a capital letter P (to show the permutation), L (to show event labels), or x (to show both).

```

1 Causets with permutations (and labels) \pcausetP{4,1,2,3},
   \pcausetX{1,2,3,4}, \pcausetL{1,2,4,3}, and without any
   \pcauset{2,4,1,3}.

```

Causets with permutations (and labels) , , , and without any .

These results are also obtained by using the optional parameter, for example:

```

1 Causets with permutations (and labels) \pcauset[permutation]{4,1,2,3},
   \pcauset[permutation,labeled]{1,2,3,4}, \pcauset[labeled]{1,2,4,3},
   and without any \pcauset{2,4,1,3}.

```

The optional argument of the main commands accepts any causets or TikZ keys to change the style of the causet. For example, the baseline can be moved with

```
1 \pcauset[baseline=2mm]{2,4,5,1,3}
```

such that the diagram is placed with its center at 2mm below the text base,




There are a few short hand macros for small common causets as shown here.

```
1 \causetFence{1}, \causetFence{2}, \ldots, \causetFence{10},
2 \causetClosedFence{1}, \causetClosedFence{2}, \ldots,
   \causetClosedFence{10},
3 \causetCrown
```



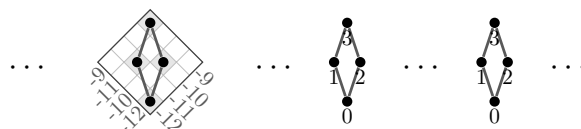
4 Labeling and replacing events (for infinite causets)

With version v1.3 of the package, it is possible to relabel and replace causet events. These features are useful for labelling events arbitrarily, and drawing diagrams of infinite causets using ellipsis.

By default, all events get named and labeled by the integers given in the arguments of the commands. For example, `\pcausetL{0,2,1}` creates the diagram  where the event notes are named (E0), (E2), (E1), respectively, while the label nodes have the names (EL0), (EL2), (EL1). If the permutation is displayed as well, the *u*- and *v*-coordinate labels are the nodes (EUL $\langle value \rangle$) and (EVL $\langle value \rangle$), with $\langle value \rangle$ taking on the values {0, 2, 1}.

To offset the names of all causet events by the same amount, use the key-value pair `offset= $\langle value \rangle$` to adjust the naming and labelling of events and get distinct event names, while the same integers are used in the command arguments. Consider the following example.

```
1 \begin{align*}
2   \tikzcausetsset{large, labels={below=1pt, at=south}}
3   \cdots\quad
4   \pcausetP[offset=-12]{0,2,1,3}\quad\cdots\quad
5   \pcausetL{0,2,1,3}\quad\cdots\quad
6   \pcausetL[offset=12]{0,2,1,3}\quad\cdots
7 \end{align*}
```



Here, the three commands have the same integer argument {0,2,1,3}, but the events are named with the specified offsets for the sub-causets, and the labels are adjusted accordingly. In this way, it will be possible to address the event (and label) nodes of the

sub-causets distinctively within a larger TikZ picture. For more details on how to draw causets within a TikZ picture, see [Sec. 9](#).

When you (only) want to change the labelling of events, add a slash and the new event label directly after the event integer in the first command argument. For example, the causet \mathbb{A} has six different (distinct) labellings by the integer set $\{0, 1, 2\}$,

```

1 \begin{align}
2   \text{order preserving:} \ \&\&
3   L_{\mathrm{op}} \ \&=
4   \left\{
5     \text{\pcausetL{1,0,2}},
6     \text{\pcausetL{1/0,0/1,2}}
7   \right\}
8   , \ \&\&
9   \text{not order preserving:} \ \&\&
10  L_{\mathrm{nop}} \ \&=
11  \left\{
12    \text{\pcauset{1/2,0/1,2/0}},
13    \text{\pcausetL{1,0/2,2/0}},
14    \text{\pcausetL{1/2,0,2/1}},
15    \text{\pcauset{1/0,0/2,2/1}}
16  \right\}
17  .
18 \end{align}

```

$$\text{order preserving:} \quad L_{\text{op}} = \left\{ \mathbb{A}_{01}, \mathbb{A}_{10} \right\}, \quad (2)$$

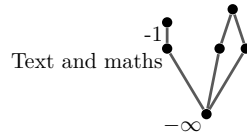
$$\text{not order preserving:} \quad L_{\text{nop}} = \left\{ \mathbb{A}_{12}, \mathbb{A}_{21}, \mathbb{A}_{02}, \mathbb{A}_{20} \right\}. \quad (3)$$

The replacing labels can be any text. It is recommended to write longer text labels in a \TeX group $\{\dots\}$. Here is an example

```

1 \pcauset[large]{2/$-\infty$,6,5,7,3/{Text and maths},4/-1}

```



Notice that replaced labels will always be visible, even if labels are deactivated (`\causetsDrawLabelsfalse`). To change the style of the replaced labels, add the key-value pair `replaced labels= $\langle value \rangle$` (which extends the style of standard labels) and set the desired style parameters in $\langle value \rangle$.

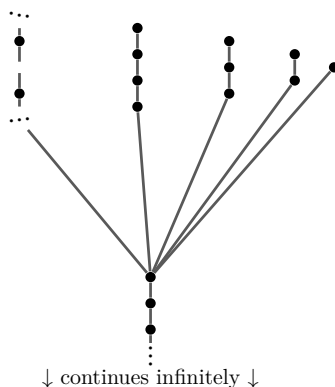
Similarly to the replacement of labels, you can replace event nodes by adding yet another slash and the event node text after the event integer in the permutation of the command. This feature may have its main application for the representation of infinite causets, for example, the code

```

1 % Define ellipsis rotated by #1 degrees:
2 \newcommand{\rdots}[1][0]{\rotatebox{#1}{...}}
3 ...
4 % Draw infinite causet with ellipsis
5 \pcauset[large, replaced labels={below, at=south}, replaced
   events={text scale=1}]
6 {-2/{\downarrow continues infinitely \downarrow}/\rdots[90],
   -1,0,1,16,14,15,11,12,13,7,8,9,10,
   2//\rdots[10],3,4//~,5,6//\rdots[-10]}

```

yields the diagram



The style of the replaced events is specified via the key `replaced labels= $\langle value \rangle$` – where $\langle value \rangle$ is `{text scale=1}` in this example. This example also shows how to relabel and replace simultaneously.

In summary, the three main commands take the form

```
\pcauset{ $\langle index \rangle$ / $\langle label \rangle$ / $\langle event text \rangle$ ,...}
```

```
\rcauset{ $\langle index \rangle$ / $\langle label \rangle$ / $\langle event text \rangle$ ,...}{ $\langle unlink from event \rangle$ / $\langle unlink to event \rangle$ ,...}
```

```
\causet{ $\langle index \rangle$ / $\langle label \rangle$ / $\langle event text \rangle$ ,...}{ $\langle link from event \rangle$ / $\langle link to event \rangle$ ,...}
```

The full notation for the short hand macros (see [Sec. 3](#)) is similar.

5 Changing sizes and fonts (for mathematical expressions)

Apart from the pre-defined sizes `tiny`, `small`, `normal`, `large` and `huge`, sizes can be specified in the optional argument of a single causet diagram or with `\tikzcausetssset{...}` for the current scope. Use the key

- `tile size= $\langle value \rangle$` to store $\langle value \rangle$ in the macro `\causetTileSize` as the size of the tiles in the permutation,
- `region line width= $\langle value \rangle$` to store $\langle value \rangle$ in the macro `\causetRegionLine` as the thickness for the line surrounding the permutation region,
- `grid line width= $\langle value \rangle$` to store $\langle value \rangle$ in the macro `\causetGridLine` as the line thickness of the permutation grid,
- `event size= $\langle value \rangle$` to store $\langle value \rangle$ in the macro `\causetEventSize` as the diameter of the events,

- `link width=\langle value \rangle` to store $\langle value \rangle$ in the macro `\causetLinkWidth` as the link thickness,
- `broken link gap=\langle value \rangle` to store $\langle value \rangle$ in the macro `\causetBrokenLinkGap` as the gap before and after an event that interrupts a link,
- `text font=\langle value \rangle` to store $\langle value \rangle$ in the macro `\causetTextFont` as the font used for all the text labels – for example, `text font=\fontsize{13}{15.6}\selectfont` sets the font size to 13pt (and the baseline skip to 1.2 times this font size), and
- `text scale=\langle value \rangle` to store $\langle value \rangle$ in the macro `\causetTextScale` as the scaling factor of all the text labels (relative to the font size).

For example, the normal size is a pre-defined style with

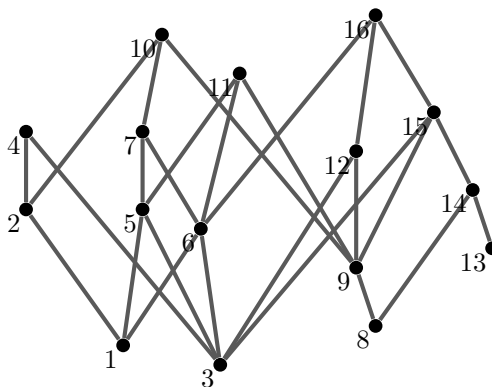
```

1 \tikzcausetsset{
2   normal/.style={
3     tile size=0.90ex,
4     region line width=0.06ex,
5     grid line width=0.03ex,
6     event size=0.48ex,
7     link width=0.15ex,
8     broken link gap=0.45ex,
9     text font=\footnotesize,
10    text scale=0.56
11  }
12 }
```

Below is an example for a diagram drawn with some sizes defined manually:

```

1 \pcausetL[tile size=2ex, event size=1ex, link width=0.3ex, text
   scale=1]{13,8,14,9,3,15,12,1,6,16,5,11,7,2,10,4}
```



Changing sizes may be required when you use causet diagrams in mathematical expression, where the size has to be adjusted for indices or limits. For example:

```

1 \begin{equation}
2   \sum_{C \in \tikzcausetsset{small}}
3   \left\{
4     \pcauset{1,2,3}, \backslash;
5     \pcauset{1,3,2,4}, \backslash;
6     \pcauset{1,4,3,2,5}, \backslash;

```

```

7      \pcauset{1,5,4,3,2,6}
8      \right\}}
9      | C |
10     = \left| \pcauset{1,2,3} \right|
11       + \left| \pcauset{1,3,2,4} \right|
12       + \left| \pcauset{1,4,3,2,5} \right|
13       + \left| \pcauset{1,5,4,3,2,6} \right|
14 \end{equation}

```

$$\sum_{C \in \left\{ \begin{array}{c} \vdots \\ \vdots \end{array}, \begin{array}{c} \vdots \\ \vdots \end{array}, \begin{array}{c} \vdots \\ \vdots \end{array}, \begin{array}{c} \vdots \\ \vdots \end{array} \right\}} |C| = \left| \begin{array}{c} \vdots \\ \vdots \end{array} \right| + \left| \begin{array}{c} \vdots \\ \vdots \end{array} \right| + \left| \begin{array}{c} \vdots \\ \vdots \end{array} \right| + \left| \begin{array}{c} \vdots \\ \vdots \end{array} \right| \quad (4)$$

6 Changing colours

Colors can be set with the following keys:

- `region colour= $\langle value \rangle$` (or `region color= $\langle value \rangle$`) for the frame of the permutation grid
- `grid colour= $\langle value \rangle$` (or `grid color= $\langle value \rangle$`) for the lines of the permutation grid
- `tile colour= $\langle value \rangle$` (or `tile color= $\langle value \rangle$`) for the occupied tiles
- `ulabel colour= $\langle value \rangle$` (or `ulabel color= $\langle value \rangle$`) for the u labels
- `vlabel colour= $\langle value \rangle$` (or `vlabel color= $\langle value \rangle$`) for the v labels
- `event colour= $\langle value \rangle$` (or `event color= $\langle value \rangle$`) for the events
- `label colour= $\langle value \rangle$` (or `label color= $\langle value \rangle$`) for the event labels
- `link colour= $\langle value \rangle$` (or `link color= $\langle value \rangle$`) for the links
- `spatial link colour= $\langle value \rangle$` (or `spatial link color= $\langle value \rangle$`) for the spatial links.

When using any of these key-value pairs in an optional arguments of the causet macros, they apply to that diagram. To change colours within a scope or the entire document, use the `\tikzcausetsset` macro, for example

```
1 \tikzcausetsset{event colour=purple, link colour=cyan}
```

changes the event colour to purple and the link colour to cyan.

For the pre-defined colour schemes, use the keys `grey colours` (or `gray colors`, default), `black colours` (or `black colors`, only black and white), `blue colours` (or `blue colors`, blue colour tones), or `neon colours` (or `neon colors`, bright colours). These keys set all the colours listed above to pre-defined values.

7 Standalone causets

For any document that contains many (large) causet diagrams or the same causets that are used repeatedly, the compilation time can be reduced by putting the commands in external files using the `standalone` package.

For example, create a file named `mycauset.tex` in the same directory as the main file with the following code:

```

1 \documentclass[tikz]{standalone}
2 \usepackage{causets}
3 \begin{document}
4 \pcauset{3,2,4,1}
5 \end{document}

```

The compiled version is then included with the macro `\causetfile{mycauset}`. The macro `\causetfile[#1]{#2}` also accepts one optional argument `[#1]` and redirects to

```

1 \includegraphics[#1]{\causetspath/#2}

```

The macro `\causetspath` is pre-defined to `.` (the directory of the main `TEX` file), but may be renewed to any desired directory with

```

1 \renewcommand{\causetspath}{my_causet_directory}

```


8 Style changes in detail

In the optional argument of the causet macros, you may use other TikZ options to change the output of the macros, for example:

```

1 \usepackage{causets}
2 ... \begin{document} ...
3 A labeled causet with red-framed semi-transparent labels,
   \pcausetL[labels={draw=red, fill=white, opacity=0.8, inner sep=1pt,
   scale=1.5}]{2,4,5,1,3}.

```

A labeled causet with red-framed semi-transparent labels, .

In the same way, it is possible to modify the existing styles of all parts of the diagrams, which means that the default drawing styles are extended by the options passed as value to these keys:

1. `region=<value>` shows the permutation (`show permutation`) and appends to the style of the permutation frame,
2. `grid=<value>` shows the permutation (`show permutation`) and appends to the style of the permutation grid,
3. `tiles=<value>` shows the permutation (`show permutation`) and appends to the style of the occupied permutation tiles,
4. `events=<value>` appends to the style of the event nodes (graph vertices),
5. `links=<value>` shows the links (`show links`) and appends to the style of the links (main graph edges),
6. `link starts=<value>` sets the starting line tip of each link,
7. `link ends=<value>` sets the ending line tip of each link,
8. `link pauses=<value>` sets the ending line tip of each link at an interruption,
9. `link resumes=<value>` sets the starting line tip of each link at an interruption,
10. `cap links` sets the starting and ending line tip of each link to a (reversed) cap,
11. `arrow links` sets the ending line tip of each link to a rounded, open, stealth arrow,

12. `spatial links=<value>` shows the spatial links (`show spatial links`) and appends to the style of the spatial links (secondary graph edges),
13. `all labels=<value>` appends to the general style of all text labels (applies to the following three label types),
14. `labels=<value>` shows the event labels (`show labels`) and appends to the style that applies only to the event labels,
15. `ulabels=<value>` shows the u labels (`show ulabels`) and appends to the style that applies only to the u labels, and
16. `vlabels=<value>` shows the v labels (`show vlabels`) and appends to the style that applies only to the v labels.

In order to manually define the full style of the parts in the diagram - not extending, but *overwriting all* drawing options - set the style of the following keys:


1. `every causet` for the general style of the causet diagram (like `baseline`),
2. `every tile` for the occupied permutation tiles,
3. `every event` for the event nodes (graph vertices),
4. `every grid` for the permutation grid,
5. `every region` for the permutation region,
6. `every label base` for all text labels,
7. `every ulabel` for the permutation u labels,
8. `every vlabel` for the permutation v labels,
9. `every link` for the links (graph edges),
10. `every spatial link` for the spacelike separations, and
11. `every label` for the event labels.

With the macro `\tikzcausetssset{...}`, any of these styles can also be set in the preamble of the document. So, for example,

```

1 \usepackage{causets}
2 \tikzcausetssset{events={rectangle}, links={-Triangle Cap[]}}
3 ... \begin{document} ...
4 \pcauset{2,4,5,1,3}

```

will generate all causets with square nodes as events and triangle caps on the future link ends, like .

9 Using this package in TikZ pictures

Since the package is TikZ based, using a causet within a TikZ picture is simple. Each of the three main commands is based on a draw command in TikZ that has the same name but starts with `\draw...`. The `\draw` macros, however, do not support an optional argument (since this can be specified directly in a TikZ picture) and they also do not have any short hand macros ending in P, L or X.

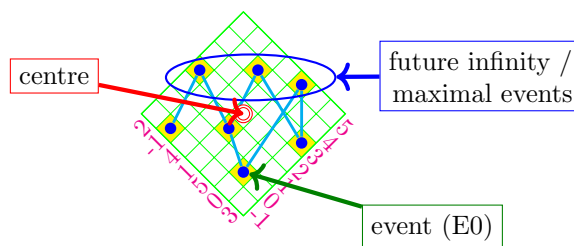
The two following examples are given by external (standalone) TikZ graphics and included with `\causetfile{...}`. More details on using causets in standalone files are given in [Sec. 7](#).

Note that the output of the `\draw` macros has the point $(0,0)$ in the center, so in order to shift it, we may enclose it in a TikZ scope, for example:

```

1 \documentclass[tikz]{standalone}
2 \usepackage[permutation,huge,neon]{causets}
3 \usetikzlibrary{fit,shapes.geometric}
4 \begin{document}
5 \begin{tikzpicture}
6   % Create causet, shifted in x- and y-direction:
7   \begin{scope}[xshift=3cm, yshift=-1cm]
8     \tikzcausetsset{offset=-2}
9     \drawpcauset{5,2,7,3,6,1,4}
10    \draw[red, double] (0, 0) circle[radius=0.1];
11  \end{scope}
12  % Circumscribe the future infinity:
13  \node[draw=blue, inner sep=1pt, thick, ellipse, fit=(E2) (E4) (E5)]
14    (Finf) {};
15  % Add labels on top:
16  \node[draw=blue, right, align=right] (FinfLabel) at (4.8, -0.5)
17    {future infinity / \\\ maximal events};
18  \node[draw=red] (centerLabel) at (0.5, -0.5) {centre};
19  \node[draw=green!50!black] (myEventLabel) at (5.5, -2.5) {event (E0)};
20  % Draw arrows from the labels to the references:
21  \draw[ultra thick, blue, ->] (FinfLabel) -- (Finf);
22  \draw[ultra thick, red, ->] (centerLabel) -- (3, -1);
23  \draw[ultra thick, green!50!black, ->] (myEventLabel) -- (E0);
24 \end{tikzpicture}
25 \end{document}

```



Note that this example also shows how to use the event nodes that are generated by the `\drawpcauset` macro. The green arrow points to the node of the second causet event (E5). The causet events, event labels, u -axis labels, and v -axis labels are the nodes (E_i) , (EL_i) , (EUL_i) , and (EVL_i) , respectively, where $i \in [1, N]$ for a causet with N events.

It is also possible to use multiple causets in more complicated graphics, like a tree for example, but note that whenever a causet is actually a text of a node within a TikZ picture (like in the following), we need to use the `\pcauset` (`\rcauset` or `\causet`) commands without the `draw` prefix again:

```

1 \documentclass[tikz]{standalone}
2 \usepackage[blue]{causets}
3 \usetikzlibrary{fit,shapes.geometric}
4 % Set colours and style:
5 \colorlet{future colour}{green!50!black}
6 \colorlet{past colour}{yellow!50!orange!80!black}
7 \colorlet{anywhere colour}{red!80!black}
8 \tikzset{prob arrow/.style={line width=1.5pt}}
9 % Define some helper functions:
10 \newcommand*{\semiopaque}[1]{% Adds a transparency group.
11 \begin{scope}[transparency group, opacity=0.5]
12   #1
13 \end{scope}
14 }
15 \newcommand*{\drawprobarrow}[4][\% Draws a (probability) arrow.
16 \draw[prob arrow] (#2) -- node[sloped, midway, below, #1] {$#3$} (#4);
17 }
18 \newcommand{\drawlegendsymbol}{% Draws the symbol for a legend entry.
19 \draw[prob arrow] (0,0) -- +(10pt, 6pt);
20 }
21 \newcommand{\legendsymbol}[1]{% Creates the legend symbol.
22 \begin{tikzpicture}
23   \ifnum#1=0
24     \drawlegendsymbol
25   \else
26     \semiopaque{\drawlegendsymbol}
27   \fi
28 \end{tikzpicture}
29 }
30 \begin{document}
31 \begin{tikzpicture}[-stealth]
32   \def\ystep{2.0cm}
33   \def\xstep{0.52cm}
34   % Create nodes of all causets with 1 to 4 events:
35   \begin{scope}[nodes={draw, thin, circle, minimum size=1.02cm}]
36     \node (C1) at ( 0, 0) {\pcauset{1}};
37     \node (C12) at (-4*\xstep, 1*\ystep) {\pcauset{1,2}};
38     \node (C21) at ( 4*\xstep, 1*\ystep) {\pcauset{2,1}};
39     \foreach \CPoint/\CXPos/\CPerm in {C123/-8/{1,2,3}},
40       C132/-4/{1,3,2}}, C312/0/{3,1,2}}, C213/4/{2,1,3}},
41       C321/8/{3,2,1}}
42     \node (\CPoint) at (\CXPos*\xstep, 2*\ystep)
43       {\expandafter\pcauset\CPerm};
44     \foreach \CPoint/\CXPos/\CPerm in {C1234/-15/{1,2,3,4}},
45       C1243/-13/{1,2,4,3}}, C1423/-11/{1,4,2,3}}, C1324/-9/{1,3,2,4}},
46       C1432/-7/{1,4,3,2}}, C4123/-5/{4,1,2,3}}, C4132/-3/{4,1,3,2}},
47       C3124/-1/{3,1,2,4}}, C3412/1/{3,4,1,2}}, C3142/3/{3,1,4,2}},
48       C4312/5/{4,3,1,2}}, C2134/7/{2,1,3,4}}, C2143/9/{2,1,4,3}},

```



```

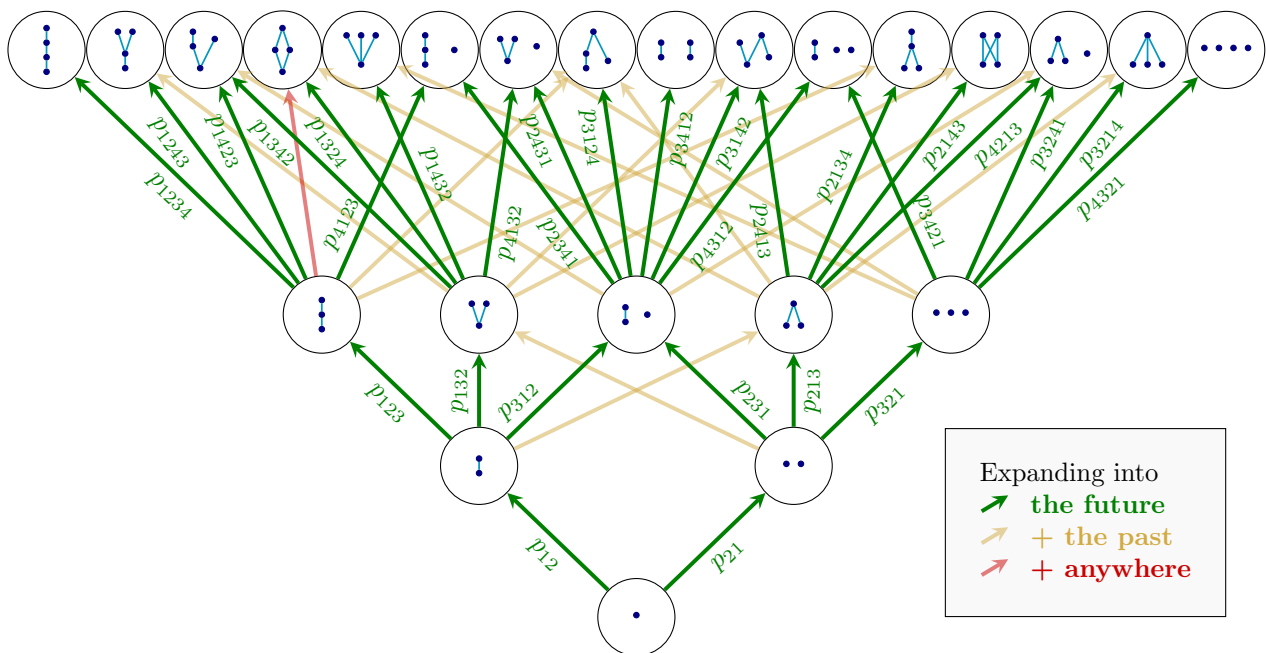
C4213/11/{4,2,1,3}}, C3214/13/{3,2,1,4}}, C4321/15/{4,3,2,1}}
42 \node (\CPoint) at (\CXPos*\xstep, 3.75*\ystep)
    {\expandafter\pcauset\CPerm};
43 \end{scope}
44 % Add graph edges for expandings ...
45 % ... to the past (additionally):
46 \begin{scope}[past colour]
47 \foreach \StartPoint/\EndPoint in {C12/C213, C123/C3124,
    C123/C2134, C132/C1243, C132/C3142, C132/C2143, C312/C1423,
    C312/C4213, C213/C1324, C213/C3124, C213/C3214, C321/C1432,
    C321/C4132, C21/C132}
48 \semiopaque{\drawprobarrow{\StartPoint}{\EndPoint}};
49 \end{scope}
50 % ... to anywhere (additionally, further):
51 \begin{scope}[anywhere colour]
52 \semiopaque{\drawprobarrow{C123}{C1324}};
53 \end{scope}
54 % ... to the future:
55 \begin{scope}[prob arrow, future colour]
56 \drawprobarrow{C1}{p_{12}}{C12}
57 \drawprobarrow{C12}{p_{123}}{C123}
58 \drawprobarrow{C123}{p_{1234}}{C1234}
59 \drawprobarrow[near end]{C123}{p_{1243}}{C1243}
60 \drawprobarrow[near end]{C123}{p_{1423}}{C1423}
61 \drawprobarrow[near start, above]{C123}{p_{4123}}{C4123}
62 \drawprobarrow[above]{C12}{p_{132}}{C132}
63 \drawprobarrow[near end]{C132}{p_{1342}}{C1423}
64 \drawprobarrow[near end]{C132}{p_{1324}}{C1324}
65 \drawprobarrow[midway, above]{C132}{p_{1432}}{C1432}
66 \drawprobarrow[near start]{C132}{p_{4132}}{C4132}
67 \drawprobarrow[near start, above]{C12}{p_{312}}{C312}
68 \drawprobarrow[near end]{C312}{p_{3124}}{C3124}
69 \drawprobarrow[near end]{C312}{p_{3412}}{C3412}
70 \drawprobarrow[near end]{C312}{p_{3142}}{C3142}
71 \drawprobarrow[near start]{C312}{p_{4312}}{C4312}
72 \drawprobarrow{C1}{p_{21}}{C21}
73 \drawprobarrow[near start, above]{C21}{p_{231}}{C312}
74 \drawprobarrow[near start]{C312}{p_{2341}}{C4123}
75 \drawprobarrow[near end]{C312}{p_{2431}}{C4132}
76 \drawprobarrow{C21}{p_{213}}{C213}
77 \drawprobarrow[above]{C213}{p_{2134}}{C2134}
78 \drawprobarrow[near start]{C213}{p_{2413}}{C3142}
79 \drawprobarrow[near end]{C213}{p_{2143}}{C2143}
80 \drawprobarrow[near end]{C213}{p_{4213}}{C4213}
81 \drawprobarrow{C21}{p_{321}}{C321}
82 \drawprobarrow[near start, above]{C321}{p_{3421}}{C4312}
83 \drawprobarrow[near end]{C321}{p_{3241}}{C4213}
84 \drawprobarrow[near end]{C321}{p_{3214}}{C3214}

```

```

85 \drawprobarrow{C321}{p_{4321}}{C4321}
86 \end{scope}
87 % Add legend:
88 \node[above left, align=left, draw, inner sep=3ex, fill=gray!05]
89   at (15*\xstep, 0)
90   {Expanding into\\
91     \bfseries\color{future colour}\legendsymbol{0}\ the future\\
92     \bfseries\color{past colour}\legendsymbol{1}\ + the past\\
93     \bfseries\color{anywhere colour}\legendsymbol{1}\ + anywhere};
94 \end{tikzpicture}
95 \end{document}

```



10 Release notes

v1.3, (planned updates)

- The numbers for the arguments of the causet commands may now be chosen from an interval $[a, b]$, where a and b do no longer have to be natural numbers but are any integers ($a \leq b$) including 0 and negative numbers to name and label the events of the causet.
- The integers given in the arguments of the causet commands that name (and auto label) events may now be modified with `offset=<value>` (0 by default) before the events are named and labeled with the given numbers.
- For all labels in the diagram, the font size is now adjustable with `text font=<value>` (additionally to `text scale=<value>`), for example to set the font size to 20pt (and a baseline skip of 24pt), use

```
1 text font=\fontsize{20}{24}\selectfont
```

- Event labels and the event nodes itself may now be replaced by user-defined text in

order to label events manually or use symbols like ... for representations of infinite causets.

- The short hand macros `*causetL...` now deactivate the permutation and `*causetP...` now deactivate labels (if activated previously).

v1.2, 2022-02-07

- The colour option `blue` has been modified so that a blue diagram is now only coloured in different shades of blue, gray and black.
- Every event label is now positioned relative to the center point of an event, so that the position of event labels can be modified with `left=<value>`, `right=<value>`, etc.
- Arrow tips may now be added to the start and end of links via the style settings `link starts=<value>` and `link ends=<value>`, or via the pre-defined styles `cap links` and `arrow links`.
- By default, links drawn by the command `\causet` are now broken if they cross over an event. The style of the start and end of the link interruption are set with `link pauses=<value>` and `link resumes=<value>`.
- Links between spacelike separated events are now also supported by the command `\rcauset`.

v1.1, 2020-12-15

- All colours, size and style options have been moved to PGF keys in the path

```
1 /tikz/causets/
```

v1.0, 2020-11-08

- First release of the package `causets` on CTAN, <https://ctan.org/pkg/causets>, including all the main commands.

11 Bug reports and package requests

Problems with the package are reported here:

<https://github.com/c-minz/LaTeX-causets/issues>

If you have a problem when using the package or you would like to have another feature to be implemented, please write a message to `christoph(dot)minz(AT)gmail(dot)com`

For reporting a bug, I much appreciated if you do the following:

- Check the issues reported previously if your problem is already listed.
- Describe your problem including the errors and warning messages, information about the drivers and programs versions.
- Provide a minimal working test file (only a single `TEX` file) that demonstrates the problem.