package

causets

version 1.5

Christoph Minz

2025/02/24

— supported with the PrOSET editor online —

$$\left\{ \cdots, \vcenter{\hbox{}}, \bigwedge, \bigvee, \vcenter{\hbox{}} \right\}, \quad \left\{ {}_1\bullet{}_2\bullet{}_3\bullet, {}_2{}_1, {}_3\bullet, {}_1{}_2, {}_2{}_3, {}_3{}_2{}_1 \right\},$$

$$\left\{ \diamondsuit, \diamondsuit, \diamondsuit, \diamondsuit, \diamondsuit \right\}$$

*The purpose of this package is to typeset partial ordered sets (posets for short) as Hasse diagrams in a style that is particularly suited for causal set theory (a theory of quantum gravity). Causal sets (**causets** for short) are locally finite, partially ordered sets, where the partial order is the causal relation between elements. This relation forms a cone structure (light-cones) such that all past and future related elements of an element fall within a lower and upper cone, respectively, opening in 90 degree angles.*

The drawing algorithm (based on TikZ) makes use of the fact that every finite causet that embeds in $1 + 1$ dimensional Minkowski spacetime is a product of two total orders (2-order), where one order is taken to be the standard total order of integers labelling all causet elements, while the other is a permutation of the labels[1]. Such a permutation is the main input for the macros. For any higher dimensional poset, there exists a 2-order such that the higher dimensional diagram is obtained by removing links from the 2-order. The online tool "PrOSET editor" helps to construct any Hasse diagram and export its macro for use in a LaTeX document.

---

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

1 The three 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 removes (restyles) 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 2-order). Let the causet 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,
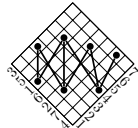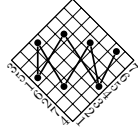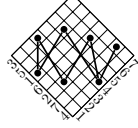
#	command	inline output
1	`\pcauset{4,7,2,6,1,5,3}`	
2	`\rcauset{4,7,2,6,1,5,3}{2/5}`	
3	`\causet{4,7,2,6,1,5,3}{1/3,1/5,2/3,2/6,4/5,4/6,4/7}`	

Table 1: Examples of the three main commands.

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 \begin{document}
4 ...
5 \begin{align}
6   S &=
7   \left\{
8     \pcauset{3,2,1}, \pcauset{3,1,2}, \pcauset{2,1,3}, \pcauset{1,3,2},
     \pcauset{1,2,3}
9   \right\}.
10 \end{align}
```

$$S = \left\{ \cdots, \vdots\, \cdot, \bigwedge, \bigvee, \vdots \right\}. \tag{1}$$

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}`, $\vdots\, \cdot$, as opposed to `\pcauset{2,3,1}`, $\cdot\, \vdots$.

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,...,5}`) or antichains like $\cdots\cdots$ (with `\pcauset{5,...,1}`). As a more complex example for the dots notation, consider

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

which produces the same diagram as

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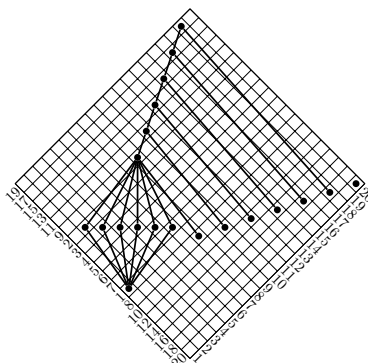

In an optional argument, the main commands accepts any causets or TikZ keys to change the style of the causet. For example, the blue causet is obtained with

```
1  \pcauset[blue colors]{2,4,5,1,3}
```

Example diagrams together with their LaTeX macros are included on the project website https://c-minz.github.io/projects/poset_diagrams. To help create and modify poset diagrams visually, I developed the PrOSET editor as an online tool, which exports the result as LaTeX macros (and other data formats):

https://c-minz.github.io/assets/html/proset-editor.html

Whenever you use this package to compose a publication to be uploaded to the arXiv, or a document that may be converted to XML or HTML with LaTeXML, this package provides special support. To replace the diagram figures especially for screen readers, add a short description with the key alt={⟨*value*⟩} in the optional argument. For example, the code

```
1  \pcauset[alt={N}]{2,4,1,3},
2  \pcauset[alt={union of an N and a 2-chain}]{5,6,2,4,1,3}
```

adds "N" as alternative text for , and "union of an N and a 2-chain" as description for . If no alternative text is specified, the default value "causet" is used (which is set in the style every causet).

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).
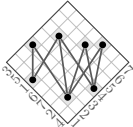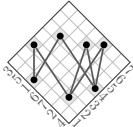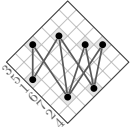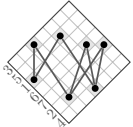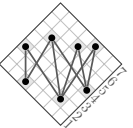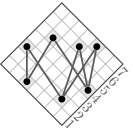
If no options are used, an output like is obtained with the code

```
1  \usepackage{causets}
2  ... \begin{document} ...
3  \rcauset[alt={3-crown with doubled top right}]{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	\pcauset	\rcauset
\usepackage[permutation]{causets} shows permutations (\causetsDrawPermutationtrue activates this option) (\causetsDrawPermutationfalse deactivates this option)		
\usepackage[permutation, uunlabeled]{causets} shows permutations without u-labels (\causetsDrawULabelstrue activates u-labels) (\causetsDrawULabelsfalse deactivates u-labels)		
\usepackage[permutation, vunlabeled]{causets} shows permutations without v-labels (\causetsDrawVLabelstrue activates v-labels) (\causetsDrawVLabelsfalse deactivates v-labels)		
\usepackage[labeled]{causets} shows labeled events (\causetsDrawLabelstrue activates labels) (\causetsDrawLabelsfalse deactivates labels)		
\usepackage[continuous]{causets} allows links to cross over events. Link interruptions (default) only affect the \causet command and "restyled" links of \rcauset, but not the examples shown on the right. Details on interrupted (broken) links are given below this table.		
\usepackage[caps]{causets} sets caps as link tips		
\usepackage[arrows]{causets} sets arrows (rounded stealth) as link tips		
\usepackage[unlinked]{causets} does not draw links (\causetsDrawLinkstrue actives links) (\causetsDrawLinksfalse deactivates links)		

continued on the next page

Package option	\pcauset	\rcauset
\usepackage[tiny]{causets} for tiny diagrams (\tikzcausetsset{tiny} sets tiny sizes)		
\usepackage[smaller]{causets} for very small diagrams (\tikzcausetsset{very small} sets very small sizes)		
\usepackage[small]{causets} for small diagrams (\tikzcausetsset{small} sets small sizes)		
For comparison, here are the normal sized causets. (\tikzcausetsset{normal} sets normal sizes)		
\usepackage[large]{causets} for large diagrams (\tikzcausetsset{large} sets large sizes)		
\usepackage[larger]{causets} for very large diagrams (\tikzcausetsset{very large} sets very large sizes)		
\usepackage[huge]{causets} for huge diagrams (\tikzcausetsset{huge} sets huge sizes) Details on size options are given in Sec. 5.		
\usepackage[black]{causets} for black and white causets (\tikzcausetsset{black colors} activates the option)		
\usepackage[gray]{causets} for grey scale causets (default) (\tikzcausetsset{gray colors} activates the option)		
\usepackage[blue]{causets} for blue causets (\tikzcausetsset{blue colors} activates the option)		
\usepackage[neon]{causets} for brightly colored causets (\tikzcausetsset{neon colors} activates the option) Details on color options are given in Sec. 6.		

Package option	\pcauset	\rcauset
\usepackage[spatial]{causets} shows dashed lines between spatially separated events (which is a secondary partial order for the causets drawn with \pcauset). This option only affects the \pcauset and \rcauset commands, but not the \causet command. (\causetsDrawSpatialLinkstrue activates the option) (\causetsDrawSpatialLinksfalse deactivates the option)		
\usepackage[external]{causets} loads the external library, sets the external file name prefix to "causets/", and enables the use of causet names for the external file names. (This feature has to be loaded as option in the preamble not later in the document.) For details on externalization of causets, see Sec. 8.		

By default, the links in a diagram are interrupted if they cross over another event (exactly over the event centre). Any such crossing of links over events can only occur for the links explicitly specified for the \causet command or the locally restyled links of the \rcauset command, but not for the links drawn by \pcauset or automatically drawn links by \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 centre that is linked to events -3 and 3 but not to events -6 and 6, while a link (blue) from -6 to 6 crosses over it.

```
1 \usepackage[larger]{causets}
2 \tikzcausetsset{labels={right, at=east}}
3 ... \begin{document} ...
4 \causet[brokenly linked]{-1,2,-5,-7,4,-3,-6,0,6,3,-4,7,5,-2,1}
     {-7/-6,-7/4,-6/-4,-6/6/blue!50!cyan,-5/-4,-5/-3,-4/-2,-4/7,
     -3/-2,-3/0,-2/1,-1/0,-1/2,0/1,0/3,2/3,2/4,3/5,4/5,4/6,6/7}
5 \causet[continuously linked]{-1,2,-5,-7,4,-3,-6,0,6,3,-4,7,5,-2,1}
     {-7/-6,-7/4,-6/-4,-6/6/blue!50!cyan,-5/-4,-5/-3,-4/-2,-4/7,
     -3/-2,-3/0,-2/1,-1/0,-1/2,0/1,0/3,2/3,2/4,3/5,4/5,4/6,6/7}
```
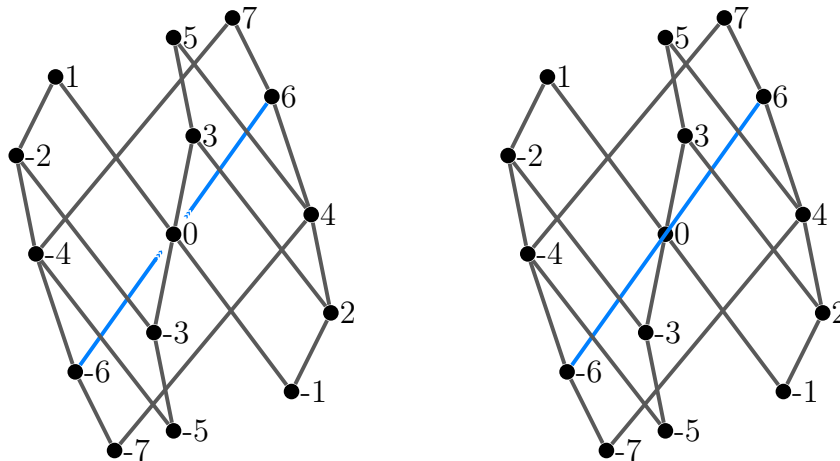
Note that the links are drawn in the order given by the second argument. Similar results are obtained with the following `\rcauset` commands (since version 1.4)

```
1 \rcauset[brokenly linked]{-1,2,-5,-7,4,-3,-6,0,6,3,-4,7,5,-2,1}
    {-7/-3,-6/0,-5/4,-4/5,0/6,3/7,-6/6/blue!50!cyan}
2 \rcauset[continuously linked]{-1,2,-5,-7,4,-3,-6,0,6,3,-4,7,5,-2,1}
    {-7/-3,-6/0,-5/4,-4/5,0/6,3/7,-6/6/blue!50!cyan}
```

where all pairs of integers in the second argument are links to be removed (by default, get the local style `draw=none`). All links with a local style specified (here the last link only) are drawn in the order they appear after all automatic links have been drawn. Local styles are explained in Sec. 7 in more detail.

All main commands, the short hand macros (see Sec. 3), and the command `\causetfile` (see Sec. 8) add a small padding to the left and right of the diagram for a better spacing within text and mathematical expressions. The padding is adjustable with `padding=`⟨*value*⟩ and stored in the macro `\causets@Padding` (by default, it is `\thinspace`).

3   Changing the options for individual diagrams

Most of the package options introduced in the section before are also defined as keys that can be specified in the optional argument of the three main macros `\pcauset`, `\rcauset` and `\causet`.

Additionally, the package has short hand macros to switch on labels, the permutation grid, or both (`\causetsDrawLabelstrue`, `\causetsDrawPermutationtrue`, or `\tikzcausetsset{ labeled, show permutation}`) to help compile diagrams. Since package version 1.5, these short hand macros ending in L, P and X are depreciated, because diagrams can now be designed and exported to LaTeX macros with the PrOSET editor. To use these short hand macros nevertheless, add the following lines to the preamble after importing the package with `\usepackage{causets}`:

```
1 \providecommand*{\pcausetP}[2][]{%
2 {\causetsDrawPermutationtrue\causetsDrawLabelsfalse
3 \pcauset[#1]{#2}}}
4 \providecommand*{\pcausetL}[2][]{%
```

```
 5 {\causetsDrawPermutationfalse\causetsDrawLabelstrue
 6 \pcauset[#1]{#2}}}
 7 \providecommand*{\pcausetX}[2][]{%
 8 {\causetsDrawPermutationtrue\causetsDrawLabelstrue
 9 \pcauset[#1]{#2}}}
10 \providecommand*{\causetP}[3][]{%
11 {\causetsDrawPermutationtrue\causetsDrawLabelsfalse
12 \causet[#1]{#2}{#3}}}
13 \providecommand*{\causetL}[3][]{%
14 {\causetsDrawPermutationfalse\causetsDrawLabelstrue
15 \causet[#1]{#2}{#3}}}
16 \providecommand*{\causetX}[3][]{%
17 {\causetsDrawPermutationtrue\causetsDrawLabelstrue
18 \causet[#1]{#2}{#3}}}
19 \providecommand*{\rcausetP}[3][]{%
20 {\causetsDrawPermutationtrue\causetsDrawLabelsfalse
21 \rcauset[#1]{#2}{#3}}}
22 \providecommand*{\rcausetL}[3][]{%
23 {\causetsDrawPermutationfalse\causetsDrawLabelstrue
24 \rcauset[#1]{#2}{#3}}}
25 \providecommand*{\rcausetX}[3][]{%
26 {\causetsDrawPermutationtrue\causetsDrawLabelstrue
27 \rcauset[#1]{#2}{#3}}}
```

To add labels or the permutation grid to a diagram, specify the respective option in the optional argument of the macro:

command with options	functionality
`\pcauset[labeled]{1,2,4,3}`	Draws a diagram of a 2-order from a permutation (here 1,2,4,3) and shows labels for the elements (`\causetsDrawLabelstrue`). The output for this example is $\overset{3}{\underset{1}{\overset{4}{\underset{2}{\mathsf{Y}}}}}$.
`\rcauset[labeled]{...}{...}`	Similar for a 2-order with removed links.
`\causet[labeled]{...}{...}`	Similar for a generic diagram.
`\pcauset[permutation]{4,1,2,3}`	Draws a diagram of a 2-order from a permutation (here 4,1,2,3) and shows the permutation grid (`\causetsDrawPermutationtrue`). The output for this example is .
`\rcauset[permutation]{...}{...}`	Similar for a 2-order with removed links.
`\causet[permutation]{...}{...}`	Similar for a generic diagram.
These options can be combined (also with other style keys, see below).	

Table 3: Main commands with options.

The short hand macros \causetFence and \causetCrown have been removed in version 1.5, since diagrams of fences, crowns, polygons and other posets are supported by the PrOSET editor and can be exported as LaTeX macros.

4 Labeling and replacing events (also to represent infinite causets)

Since 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, \pcauset[labeled]{0,2,1} creates the diagram V where the event notes are named (E0), (E2), (E1) (which are always created), respectively, while the label nodes have the names (EL0), (EL2), (EL1) (which are only created with the labeled option). If the permutation is displayed as well, the u- and v-coordinate labels are the nodes (EUL⟨value⟩) and (EVL⟨value⟩), with ⟨value⟩ taking on the values $\{0, 2, 1\}$.

To offset the names of all causet events by an integer amount, use the key-value pair offset=⟨value⟩ 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}
3   \cdots\quad
4   \pcauset[permutation, offset=-12]{0,2,1,3}\quad\cdots\quad
5   \pcauset[labeled]{0,2,1,3}\quad\cdots\quad
6   \pcauset[labeled, offset=12]{0,2,1,3}\quad\cdots
7 \end{align*}
```
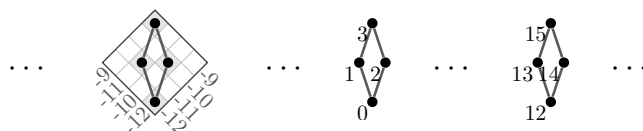

Here, the three commands have the same integer argument {0,2,1,3}, but the events are named with the specified offsets and the labels are adjusted accordingly. In this way, it will be possible to address the event (and label) nodes of the (sub-)causets distinctively within a larger TikZ picture. For more details on how to draw causets within a TikZ picture, see Sec. 10.

When you (only) want to change the labelling of an event to ⟨value⟩, add a slash and label=⟨value⟩ directly after the event integer in the first command argument. For example, the causet Λ has six different labellings by (distinct) integers from the set $\{0, 1, 2\}$,

```
1 \begin{align}
2   \text{order-preserving:} &&
3   L_{\mathrm{op}} &=
4   \left\{
5     \pcauset[labeled]{1,0,2},
```

```
6      \pcauset[labeled]{1/label=0,0/label=1,2}
7    \right\}
8    , \\
9    \text{not order-preserving:} &&
10   L_{\mathrm{nop}} &=
11   \left\{
12     \pcauset{1/label=2,0/label=1,2/label=0},
13     \pcauset[labeled]{1,0/label=2,2/label=0},
14     \pcauset[labeled]{1/label=2,0,2/label=1},
15     \pcauset{1/label=0,0/label=2,2/label=1}
16   \right\}
17   .
18 \end{align}
```

order-preserving:

$$L_{\mathrm{op}} = \left\{ \underset{0\ 1}{\overset{2}{\bigwedge}}, \underset{1\ 0}{\overset{2}{\bigwedge}} \right\}, \tag{2}$$

not order-preserving:

$$L_{\mathrm{nop}} = \left\{ \underset{1\ 2}{\overset{0}{\bigwedge}}, \underset{2\ 1}{\overset{0}{\bigwedge}}, \underset{0\ 2}{\overset{1}{\bigwedge}}, \underset{2\ 0}{\overset{1}{\bigwedge}} \right\}. \tag{3}$$

You may use any text as a label. Longer text labels should be written in a TeX group {...}. Here is an example

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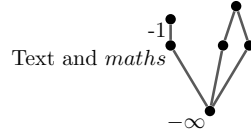

Notice that replaced labels will always be visible, even if labels are deactivated (\causetsDrawLabelsfalse) as in this example. To change the style of the replaced labels, add the key-value pair `replaced labels=⟨value⟩`, which extends the style of standard labels.

Similarly to the replacement of labels, you can replace event nodes by using a slash followed by `event=⟨value⟩`. 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/{label={$\downarrow$ continues infinitely $\downarrow$},
    event={\rdots[90]}}, -1,0,1,16,14,15,11,12,13,7,8,9,10,
    2/event={\rdots[10]},3,4/event={~},5,6/event={\rdots[-10]}}
```

yields the diagram

↓ continues infinitely ↓
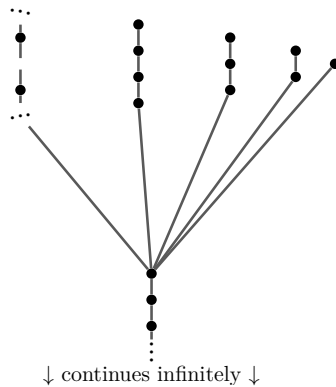
Here, three events are replaced by ellipsis with different rotations, and one event is replaced by an empty node (`˜`). The style of the replaced events is specified via the key `replaced labels=`⟨*value*⟩ – where ⟨*value*⟩ is `{text scale=1}` in this example. This example also shows how to relabel and replace simultaneously (see the bottom-most event).

The style and the position of all event labels is set with `labels=`⟨*value*⟩, while `replaced labels=`⟨*value*⟩ extends this style. As shown in the previous example, the position of labels is defined with a key word like `left`, `right`, `above`, `below`, etc. (see also the TikZ manual) relative to the position given by the key `at=`⟨*value*⟩. The `at` key holds a coordinate of the event note. The default positioning is `below left, at=south west`.

5   Changing sizes and fonts (for mathematical expressions)

The pre-defined size styles have the following scaling with respect to the normal size:

- `tiny` is $\frac{1}{3}$ times the normal size
- `very small` is $\frac{4}{9}$ times the normal size
- `small` is $\frac{2}{3}$ times the normal size
- `large` is $\frac{3}{2}$ times the normal size
- `very large` is $\frac{9}{4}$ times the normal size
- `huge` is 3 times the normal size

Apart from the pre-defined sizes, a causet can be scaled to any size using `size unit=`⟨*value*⟩ (which is equal to the tile size, `tile size=`⟨*value*⟩). The normal size unit has the value `0.9ex`.

I recommend to change the size of a causet when used in indices or limits for mathematical expression. For a (first level) sub-script in a summation, you may use the small size,

```
1  \begin{equation}
2      \sum_{C \in \tikzcausetsset{small}
3          \Bigl\{ \pcauset{1,2,3}, \pcauset{1,3,2,4}, \pcauset{1,4,3,2,5},
      \pcauset{1,5,4,3,2,6} \Bigr\}} | C |
4    = \left| \pcauset{1,2,3} \right|
5      + \left| \pcauset{1,3,2,4} \right|
6      + \left| \pcauset{1,4,3,2,5} \right|
7      + \left| \pcauset{1,5,4,3,2,6} \right|
8  \end{equation}
```

$$\sum_{C \in \left\{ \begin{smallmatrix} \end{smallmatrix}, \end{smallmatrix}, \end{smallmatrix}, \end{smallmatrix} \right\}} |C| = \left| \begin{smallmatrix} \end{smallmatrix} \right| + \left| \begin{smallmatrix} \end{smallmatrix} \right| + \left| \begin{smallmatrix} \end{smallmatrix} \right| + \left| \begin{smallmatrix} \end{smallmatrix} \right| \tag{4}$$

Like any other options, sizes can also be specified in the optional argument of a single causet diagram or with `\tikzcausetsset{...}` for the current scope.

If you need to set the sizes of individual components of a causet, use the keys

- `tile size=`⟨*value*⟩ to store ⟨*value*⟩ in the macro `\causetTileSize` as the size of the tiles in the permutation (this is equivalent to setting the size unit),
- `region line width=`⟨*value*⟩ to set the thickness for the line surrounding the permutation region,
- `grid line width=`⟨*value*⟩ to set the line thickness of the permutation grid,
- `event size=`⟨*value*⟩ to set the diameter of the events,
- `event outline=`⟨*value*⟩ to set the thickness of the transparent line around the events,
- `link width=`⟨*value*⟩ to set the link thickness, and
- `broken link gap=`⟨*value*⟩ to set the link gap before and after an event that interrupts a link.

The text of all labels is sized independently to the other components of the causet diagrams. To adjust the text typesetting, use

- `text font=`⟨*value*⟩ to set 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),
- `text scale=`⟨*value*⟩ to set the scaling factor of all the text labels (relative to the font size).

The pre-defined size styles use the following font sizes,

- `tiny` is typeset with `\tiny` (scaled by 0.30)
- `very small` is typeset with `\tiny` (scaled by 0.40)
- `small` is typeset with `\scriptsize` (scaled by 0.43)
- `normal` is typeset with `\footnotesize` (scaled by 0.56)
- `large` is typeset with `\small` (scaled by 0.75)
- `very large` is typeset with `\normalsize` (scaled by 1.02)
- `huge` is typeset with `\Large` (scaled by 0.97)

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

```
\pcauset[labeled, 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}
```
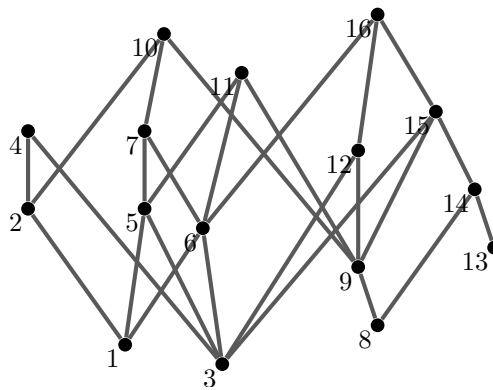
6 Changing colors

Colors can be set with the following keys:

- `region color=`⟨*value*⟩ (or `region color=`⟨*value*⟩) for the frame of the permutation grid
- `grid color=`⟨*value*⟩ (or `grid color=`⟨*value*⟩) for the lines of the permutation grid
- `tile color=`⟨*value*⟩ (or `tile color=`⟨*value*⟩) for the occupied tiles
- `ulabel color=`⟨*value*⟩ (or `ulabel color=`⟨*value*⟩) for the u labels
- `vlabel color=`⟨*value*⟩ (or `vlabel color=`⟨*value*⟩) for the v labels
- `event color=`⟨*value*⟩ (or `event color=`⟨*value*⟩) for the events
- `label color=`⟨*value*⟩ (or `label color=`⟨*value*⟩) for the event labels
- `link color=`⟨*value*⟩ (or `link color=`⟨*value*⟩) for the links
- `spatial link color=`⟨*value*⟩ (or `spatial link color=`⟨*value*⟩) 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 colors within a scope or the entire document, use the `\tikzcausetsset` macro, for example

```
\tikzcausetsset{event color=purple, link color=cyan}
```

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

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

7 Changing the styles of individual events and links

The full supported formats of the three main commands are

`\pcauset[`⟨*causet style*⟩`]{`⟨*index*⟩`/`⟨*event style*⟩`,...}`

`\rcauset[`⟨*causet style*⟩`]{`⟨*index*⟩`/`⟨*event style*⟩`,...}{`⟨*unlink from*⟩`/`⟨*unlink to*⟩`,...,`
⟨*link from*⟩`/`⟨*link to*⟩`/`⟨*link style*⟩`,...}`

`\causet[`⟨*causet style*⟩`]{`⟨*index*⟩`/`⟨*event style*⟩`,...}{`⟨*link from*⟩`/`⟨*link to*⟩`/`⟨*link style*⟩`,...}`

and similarly for all short hand macros as described in Sec. 3. The style values are all optional, where ⟨*causet style*⟩ is applied to that causet only, ⟨*event style*⟩ is applied to a single event, and ⟨*link style*⟩ is applied to a single link.

The feature to apply styles only to individual events and links is to highlight or modify subsets of a causet, for example, by using different colors:

```
1 \rcauset[labeled, large, subset A/.style={red, dotted}, subset
    B/.style=blue, replaced labels={text scale=1, at=east,
    right}]{2/{subset A, label=event $a$}, 5/subset B, 1, 3/{subset A,
    label=$b$, replaced labels=orange}, 4/subset B, 6/subset
    B}{2/3/subset A, 4/6/subset B, 5/6/subset B}
```
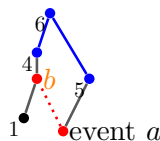

Note that `label=event $a$` and `label=$b$` adds (replaced) labels *a* and *b*, respectively, but their style is not affected by `subset A`. To change the style of the labels, we have to use the `replaced labels` key, see also Sec. 4.

8   Externalisation and 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 moving the graphics to external files using the externalisation library of TikZ or `standalone` files.

The externalisation feature of TikZ provides the tools to automatically externalise all causet diagrams and only recompile them when they are changed in the document. To enable this feature, use the `external` option when loading the package and make sure that the document is compiled with the flag

```
1 --shell-escape
```

The `external` option loads the `external` library, sets the `prefix` to add "`causets/`" before each external file, and calls `\causetsNameExternaltrue` to use the manually specified names for external file names. Any causet can be manually named by providing the name key-value pair `name=`⟨*value*⟩ in the optional parameter of a causet command. As an example, the inline code

```
1 \pcauset[name=Vee, alt={Vee}]{1,3,2}
```

yields the causet ⋁ and creates an external pdf file with the name `causets/Vee.pdf` relative to the main tex file. The alternative text `alt={`⟨*value*⟩`}` may be the same as name for short names (like here), but the file names are usually shorter.

I recommend to always name causets, even if externalisation is not used. In this way, it is easier to read the source code and switch to externalising at a later point more easily. Note that the name of a causet does not have to be unique (especially if it is used repeatedly throughout the document), but you may give different names

to causets with different style options to avoid unwanted overwrites. For a repeated causet in the document, either use the same code or the `\causetfile` macro. For example, `\pcauset[name=N]{2,4,1,3}` can be repeated with `\causetfile{N}` to print the N causet ⋀. The macro `\causetfile[#1]{#2}` also accepts one optional argument `[#1]` and expands to

```
1 \causets@Padding%
2 \includegraphics[#1]{\causets@NamePrefix#2}%
3 \causets@Padding
```

To redefine the prefix stored in `\causets@NamePrefix` and also set the `external` library prefix to the same value, use `\tikzcausetsset{set prefix=causets/}`. However, to set `\causets@NamePrefix` only for `causets` but not for other TikZ externalisation, use `name prefix=⟨value⟩`.

If you are already loading the `external` library for other uses, you do not want to use the `external` option, but add the following to the document's preamble:

```
1 \usetikzlibrary{external}
2 ...
3 \usepackage{causets}
4 \tikzcausetsset{set prefix=causets/, name external=true}
```

For externalising only those causets that are named manually, set

```
1 \tikzset{external/only named=true}
```

For switching externalisation on/off all together, the library provides the macros

```
1 \tikzexternaldisable
2 \tikzexternalenable
```

Please find further information on the `external` library in the TikZ manual.

Graphics may also be moved to standalone tex files manually and included with `\causetfile` or `\includegraphics`. For example, create a file named `M.tex` in the directory that was set to `name prefix` (by default, this is "./") with the following code:

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

The compiled diagram ⋀⋀ is then included with the macro `\causetfile[alt={M}]{M}` (where `alt` is directly passed to `\includegraphics` to be processed with LaTeXML).

More examples on larger standalone graphics are given in .

9 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,
    \pcauset[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, $\begin{smallmatrix}3&5\\1&4\\&2\end{smallmatrix}$.

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, 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`, see below),
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 `\tikzcausetsset{...}`, any of these styles can also be set in the preamble of the document. So, for example,

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

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

By default, causets are placed such that a pre-defined coordinate point falls on the text baseline, `baseline=(baseline point)`. The baseline point is `tile size`$/\sqrt{2}$ below the centre of the diagram. You may adjust the baseline as for any TikZ graphic, for example,

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

places the diagram centre 2mm below the text base, .

10   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 `\draw...` do not add extra padding (`\causets@Padding`) on the left and right of the graphic either.

The two following examples are given by external (standalone) TikZ graphics and included with `\includegraphics{...}`. More details on using causets in standalone files are given in Sec. 8.

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

```
1 \documentclass[tikz]{standalone}
2 \usepackage[permutation,larger,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  % Mark the maximal elements (future infinity):
13  \node[draw=blue, inner sep=1pt, thick, ellipse, fit=(E2) (E4) (E5)]
      (Finf) {};
14  % Add labels on top:
15  \node[draw=blue, right, align=right] (FinfLabel) at (4.8, -0.5)
      {maximal events / \\ (future infinity)};
16  \node[draw=red] (centerLabel) at (0.5, -0.5) {centre};
17  \node[draw=green!50!black] (myEventLabel) at (5.5, -2.5) {event (E0)};
18  % Draw arrows from the labels to the references:
19  \draw[ultra thick, blue, ->] (FinfLabel) -- (Finf);
20  \draw[ultra thick, red, ->] (centerLabel) -- (3, -1);  % centre was
      shifted by the scope
21  \draw[ultra thick, green!50!black, ->] (myEventLabel) -- (E0);
22 \end{tikzpicture}
23 \end{document}
```
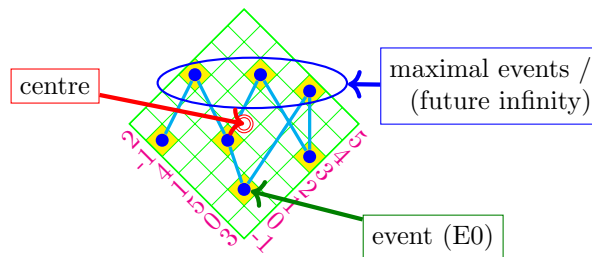

Note that this example also shows how to use the event nodes that are generated by the `\drawpcauset` macro. The green arrow points to the node of the second causet event (`E5`). The causet events, event labels, u-axis labels, and v-axis lables are the nodes (`Ei`), (`ELi`), (`EULi`), and (`EVLi`), respectively, where $i \in [-1, 5]$.

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[black]{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 \def\defCevents#1#2#3{\xdef\Ea{#1}\xdef\Eb{#2}\xdef\Ec{#3}}
11 \def\defDevents#1#2#3#4{\expandafter\defCevents#1#2#3\xdef\Ed{#4}}
12 \newcommand*{\semiopaque}[1]{% Adds a transparency group.
13 \begin{scope}[transparency group, opacity=0.5]
14   #1
15 \end{scope}
16 }
17 \newcommand*{\drawprobarrow}[4][]{% Draws a (probability) arrow.
18 \draw[prob arrow] (#2) -- node[sloped, midway, below, #1] {$#3$} (#4);
19 }
20 \newcommand{\drawlegendsymbol}{% Draws the symbol for a legend entry.
21 \draw[prob arrow] (0,0) -- +(10pt, 6pt);
22 }
23 \newcommand{\legendsymbol}[1]{% Creates the legend symbol.
24 \begin{tikzpicture}
25   \ifnum#1=0
26     \drawlegendsymbol
27   \else
28     \semiopaque{\drawlegendsymbol}
29   \fi
30 \end{tikzpicture}
31 }
32 \begin{document}
33 \begin{tikzpicture}[-stealth]
34   \def\ystep{2.0cm}
35   \def\xstep{0.60cm}
36   % Create nodes of all causets with 1 to 4 events:
37   \begin{scope}[nodes={draw, thin, circle, minimum size=1.1cm}]
38     \node (C1) at ( 0,  0) {\pcauset{1}};
39     \node (C12) at (-4*\xstep,  1*\ystep) {\pcauset{1,2}};
40     \node (C21) at ( 4*\xstep,  1*\ystep) {\pcauset{2,1}};
41     \foreach \perm [count=\i from 0] in {123, 132, 312, 213, 321}{%
42       \expandafter\defCevents\perm
43       \node (C\perm) at (4*\i*\xstep-8*\xstep, 2*\ystep)
    {\pcauset{\Ea,\Eb,\Ec}};
44      }
45     \foreach \perm [count=\i from 0] in {1234, 1243, 1423, 1324, 1432,
    4123, 4132, 3124, 3412, 3142, 4312, 2134, 2143, 4213, 3214, 4321}{%
46       \expandafter\defDevents\perm
47       \node (C\perm) at (2*\i*\xstep-15*\xstep, 3.75*\ystep)
    {\pcauset{\Ea,\Eb,\Ec,\Ed}};
48      }
49   \end{scope}
50   % Add graph edges for expandings ...
51   % ... to the past (additionally):
52   \begin{scope}[past colour]
53     \foreach \perma/\permb in {12/213, 123/3124, 123/2134, 132/1243,
    132/3142, 132/2143, 312/1423, 312/4213, 213/1324, 213/3124,
```

```
        213/3214, 321/1432, 321/4132, 21/132}
54        \semiopaque{\drawprobarrow{C\perma}{}{C\permb}};
55    \end{scope}
56    % ... to anywhere (additionally, further):
57    \begin{scope}[anywhere colour]
58        \semiopaque{\drawprobarrow{C123}{}{C1324}};
59    \end{scope}
60    % ... to the future:
61    \begin{scope}[prob arrow, future colour]
62      \drawprobarrow{C1}{p_{12}}{C12}
63      \drawprobarrow{C12}{p_{123}}{C123}
64      \drawprobarrow{C123}{p_{1234}}{C1234}
65      \drawprobarrow[near end]{C123}{p_{1243}}{C1243}
66      \drawprobarrow[near end]{C123}{p_{1423}}{C1423}
67      \drawprobarrow[near start, above]{C123}{p_{4123}}{C4123}
68      \drawprobarrow[above]{C12}{p_{132}}{C132}
69      \drawprobarrow[near end]{C132}{p_{1342}}{C1423}
70      \drawprobarrow[near end]{C132}{p_{1324}}{C1324}
71      \drawprobarrow[midway, above]{C132}{p_{1432}}{C1432}
72      \drawprobarrow[near start]{C132}{p_{4132}}{C4132}
73      \drawprobarrow[near start, above]{C12}{p_{312}}{C312}
74      \drawprobarrow[near end]{C312}{p_{3124}}{C3124}
75      \drawprobarrow[near end]{C312}{p_{3412}}{C3412}
76      \drawprobarrow[near end]{C312}{p_{3142}}{C3142}
77      \drawprobarrow[near start]{C312}{p_{4312}}{C4312}
78      \drawprobarrow{C1}{p_{21}}{C21}
79      \drawprobarrow[near start, above]{C21}{p_{231}}{C312}
80      \drawprobarrow[near start]{C312}{p_{2341}}{C4123}
81      \drawprobarrow[near end]{C312}{p_{2431}}{C4132}
82      \drawprobarrow{C21}{p_{213}}{C213}
83      \drawprobarrow[above]{C213}{p_{2134}}{C2134}
84      \drawprobarrow[near start]{C213}{p_{2413}}{C3142}
85      \drawprobarrow[near end]{C213}{p_{2143}}{C2143}
86      \drawprobarrow[near end]{C213}{p_{4213}}{C4213}
87      \drawprobarrow{C21}{p_{321}}{C321}
88      \drawprobarrow[near start, above]{C321}{p_{3421}}{C4312}
89      \drawprobarrow[near end]{C321}{p_{3241}}{C4213}
90      \drawprobarrow[near end]{C321}{p_{3214}}{C3214}
91      \drawprobarrow{C321}{p_{4321}}{C4321}
92    \end{scope}
93    % Add legend:
94    \node[above left, align=left, draw, inner sep=3ex, fill=gray!05]
95      at (15*\xstep, 0)
96      {Expanding towards\\
97      \bfseries\color{future colour}\legendsymbol{0}\ the future\\
98      \bfseries\color{past colour}\legendsymbol{1}\ + the past\\
99      \bfseries\color{anywhere colour}\legendsymbol{1}\ + anywhere};
100 \end{tikzpicture}
```
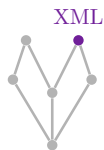
```
101  \end{document}
```


(rescaled to 80%)
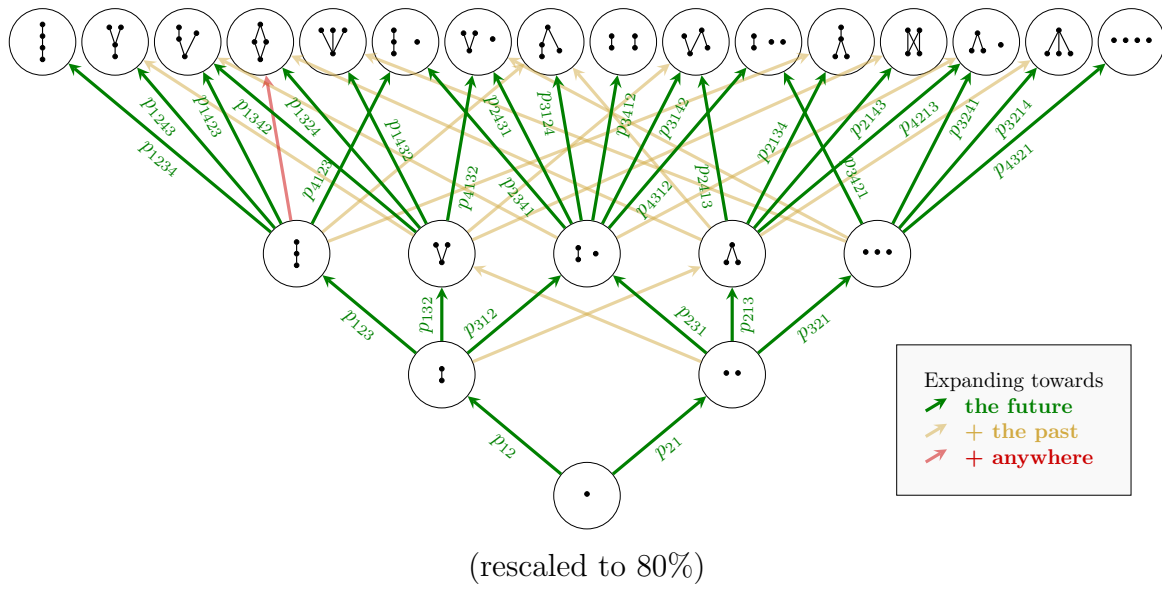
11   Release notes

v1.5, 2025-02-24

New features

- Added support for LaTeXML including the key `alt={⟨value⟩}` to define an alternative text representing a diagram for screen readers.
- Added references to a supporting online tool, the PrOSET editor.

Modifications and fixes

- The short hand macros `\pcausetL`, `\pcausetP`, `\pcausetX`, `\rcausetL`, `\rcausetP`, `\rcausetX`, `\causetL`, `\causetP` and `\causetX` are now depreciated.
  Write `\pcauset[labeled]`, `\pcauset[permutation]`, `\pcauset[permutation, labeled]`, etc. instead, or use the PrOSET editor to design diagrams.
- Removed the short hand macros `\causetFence` and `\causetCrown`.
- Corrected typos in the manual.

v1.4, 2023-03-06

New features

- Second values in the sequence of the first argument: now modifies the style of individual events, the events are replaced with the key `event=⟨value⟩`, and the labels are set with `label=⟨value⟩`.
- Third values in the sequence of the second argument of `\causet` and `\rcauset`: to modify the style of individual links.
- Broken links for restyled links of `\rcauset`: restyled links can also be drawn between events that would have not been automatically linked, hence these links might now be broken up like the links of `\causet`.
- New key `name=⟨value⟩`: to define a name of a causet for easier readability of the tex source code (and to be used as external file name when the corresponding package option has been activated).
- New option `external`: to load the `external` library of TikZ for automatic externalisation of all diagrams.
- New key `set prefix=⟨value⟩`: to pass a file name prefix that is combined with a causet name for the external files when using `external`.
- New key `name external=⟨bool⟩`: to activate/deactivate using the causet names for external files.
- Main commands: padded by a thin space (adjustable with `padding=⟨value⟩`) for a better spacing in text and math mode.

Modifications and fixes

- Third values in the sequence of the first argument: no longer supported.
- Spatial links: are now drawn before (so on a lower layer than) links.
- Style `arrow links`: the arrow heads are now filled.

- Macro `\causetspath`: has been replaced by `name prefix=`⟨*value*⟩.
- Short-hand macro `\causetCrown`: changed to take a single argument (and the optional style argument) and returns the same output as `\causetClosedFence` used to, which has been removed.

v1.3.1, 2022-07-27

Modifications and fixes

- Automatic event labels: now correctly printed when using `offset=`⟨*value*⟩.

v1.3, 2022-05-23

New features

- Integer sequences in the arguments: use a permutation of an interval $[a, b]$, where a and b do no longer have to be natural numbers but can be any integers ($a \leq b$) including 0 and negative numbers. These numbers name and label the events of the causet if the respective options are chosen.
- Second values in the sequence of the first argument: to change the name and label of events.
- New key `offset=`⟨*value*⟩ (0 by default): to define an integer shift for all events before they are named and labeled.
- New style `text font=`⟨*value*⟩ (additionally to `text scale=`⟨*value*⟩): to adjust the font style of text (labels), for example, to set the font size to 20pt (and a baseline skip of 24pt), use `text font=\fontsize{20}{24}\selectfont`

Modifications and fixes

- Size options: all sizes are now scaled with respect to the normal size.
- Event style: line thickness of the event outline has been corrected.
- Size macros removed: `\causetRegionLine`, `\causetGridLine`, `\causetEventSize`, `\causetLinkWidth`, and `\causetBrokenLinkGap`. The respective values are now stored in PGF keys and are measured with respect to the size unit (stored in `\causetTileSize`).
- All package options: reduced to single words.
- Permutation for `\*causetL...`: now deactivated (if activated previously), by default.
- Labels for `\*causetP...`: now deactivated (if activated previously), by default.

v1.2, 2022-02-07

New features

- New style `at=`⟨*value*⟩: to specify the relative point of events (`at=`⟨*value*⟩, by default) and position relative to this point using TikZ keys `left=`⟨*value*⟩, `right=`⟨*value*⟩, etc.
- New styles `cap links` and `arrow links`: add different arrow tips to the start and end of links.
- New styles `link starts=`⟨*value*⟩ and `link ends=`⟨*value*⟩: to modify arrow tips to the start and end of links.

- Broken links for `\causet`: links crossing over unlinked events are broken up now.
- New styles `link pauses`=⟨*value*⟩ and `link resumes`=⟨*value*⟩: to set the link ending style (arrows) at breaking points.
- Spatial links for `\rcauset`: spatial links for spacelike separated events are now supported by `\rcauset`, too.

Modifications and fixes
- Color option `blue`: blue colors are now different shades of blue, gray and black only.

v1.1, 2020-12-15

New features
- All colors, 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.

All previous package versions are at
https://github.com/c-minz/LaTeX-causets/tree/main/previous_versions

12   Bug reports and package requests

To see reported problems, go to 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.