# Creating diagrams for chess problems

Version v1.23

Thomas Brand
Bornheim

Stefan Höning
Neuss

2025/03/12

**Abstract**

It have been more than ten years now, since we last published a documented version of the `diagram.sty`, which is mainly intended to be used for typesetting chess problems. Since 1994 I (Stefan Höning) made a couple of enhancements to the sourcecode of the style, without publishing and putting this into the documentation. We also needed to upgrade to LaTeX 2$_\varepsilon$. The major change is the documentation language, which is english now.

The style itself tries to collect very detailed information about a chess problem by providing a lot of commands, which you may use to specify the necessary information. There are different reasons for this. One idea was to enable people to read LaTeX-diagrams into databases with information as detailed as possible. Otherwise it should be easy to change the layout of a diagram by applying a changed style - not by changing the source.

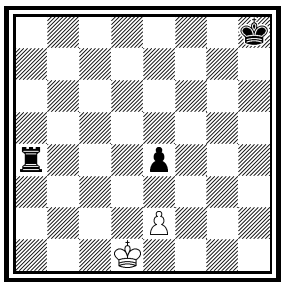
# Contents

# 1 Creating diagrams

## 1.1 An introductary example

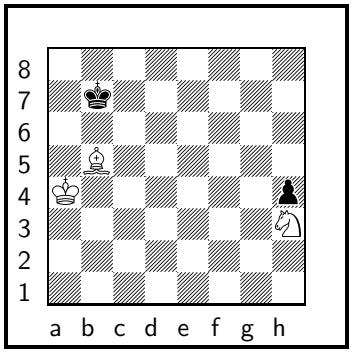
Let us first take a look at a simple example which should only show what you have to type into your LaTeX-code to get nice looking diagrams.

**1**
**Thomas Brand**
***Problemkiste 1992***
*Elmar Bartel gew.*



h#7 C- (2+3)

**2**
**Thomas Brand**
***Problemkiste 1992***




h#5 C- (3+2)

**1) Thomas Brand:**
1.Ta3 Kc2!, 2.Tf3 e×f3, 3.e3 f4, 4.e2 f5, 5.e1T f6, 6.Th1! (Te7?) f7, 7.Th7 f8D#
**2) Thomas Brand:**
1.Ka8 Sg1, 2.h3 Ka5, 3.h2 Kb6, 4.h×g1L+ Kc7, 5.La7 Lc6#

To use the package you have to make it available to LaTeX using `\usepackage{diagram}` inside the preamble of your document.

Then you may use the `diagram` environment to create the diagrams. For the above example I had to type the following:

```
 \begin{diagram}
  \author{Brand, Thomas}
  \source{Problemkiste} \year{1992}
  \dedic{Elmar Bartel gew.}
  \pieces[2+3]{wKd1, wBe2, sKh8, sBe4, sTa4}
  \stip{h\#7}
  \sol{1.Ta3 Kc2!, 2.Tf3 e\x f3, 3.e3 f4, 4.e2 f5, 5.e1T f6,
       6.Th1! (Te7?) f7, 7.Th7 f8D\#}
 \end{diagram}
%
 \hfill
%
 \begin{diagram}
  \setboolean{legend}{true}
  \author{Brand, Thomas}
  \source{Problemkiste} \year{1992}
  \pieces[3+2]{wKa4, wLb5, wSh3, sKb7, sBh4}
  \stip{h\#5}
  \sol{1.Ka8 Sg1, 2.h3 Ka5, 3.h2 Kb6, 4.h\x g1L+ Kc7, 5.La7 Lc6\#}
 \end{diagram}
```

\putsol

diagram

Any information which belongs to a problem should be put between \begin{diagram} and \end{diagram}. The above examples contains information for *authors, source, year of publication, stipulation, solution* and (in diagram 1) a *dedication*.

This information is shown around a chessboard except the solution, which is collected and put into the output using the \putsol command.

## 1.2 Elements of a diagram

This section describes the elements which may be used inside a diagram environment. For most of these elements there is no sense using them between \begin{diagram} and \end{diagram}. Some of them will not work outside of the environment (like —). In case you use these switches anywhere outside you will specify the information for all problems in your surrounding environment (which may be the complete document).

### 1.2.1 Collecting the problem information

The following information is typically given with a problem:

\author

- With the \author tag you specify one author or a list of authors. If you specify more than one author, you must separate them with "; ". Normally an author is given as *"sirname, givenname"*. You may change the way, how the name is interpreted by LaTeX using \normalnames and \reversednames. This \author command does only overwrite the default behaviour when used inside a diagram environment.

\Dr
\Prof
\ProfDr

- Within the Authors command you should use the commands \Dr, \Prof and \ProfDr to specify these academic titles. So one may switch off the display of these titles — like it is generally done inside ***Die Schwalbe***.

\pieces

- With \pieces you specify the position to be displayed on the board. For each kind of piece you may specify a list of fields. Different lists of fields are separated by ", ". So the general syntax for specifying the position of a specific piece is:
  [color][piece]{rotation of piece}[list of squares];
  e. g. wTa1h1 should be clear, nKa4 is a neutral king on a4
  **w s n** may be used to specify the color of the piece.
  **K D T L S B C E X** may be used to specify the piece. A **C** is used for an imitator, **E** for an equihopper and **X** for a rotated equihopper. You may *not* use an optional rotation with **C, E** and **X**.
  **R U L** may be used to specify an optional rotation: right, upside-down, left. So you may use sDUc7 for a grashopper on c7 — displayed as an upsidedown queen.
  The characters used to specify color, piece and rotation may be changed using the \DefinePieces command.

  You may also optionally specify the number of pieces in your diagram, which then will be used to control your input automatically.

There is also support for an imitator, which is typically displayed as a black filled circle. So `sCf4` will produce the symbol of an imitator. This is shown in diagram 3.

`\fen`
- As an alternative notation it is possible to enter the position in *Forsyth-Edwards-Notation*. This is possible for $8 \times 8$ boards only.

  As an example: The position in diagram 2 was created via `\fen{8/1k6/8/1B6/K6p/7N/8/8}`.

  As with the `\pieces` command, you may provide the number of white and black pieces as an optional parameter.

`\stipulation` `\stip`
- is used to specify the stipulation of the problem, e. g. `\stipulation{\#2}` may be used to specify a *mate in two*. There is also an abbreviation `\stip` for this macro.

`\city`
- may be used to specify the city and country, where the author or the authors live. I use this inside the original section of ***Die Schwalbe***. You should separate multiple cities (for multiple authors) with "; ". There is also a boolean switch `showcity`, which contols, whether this information is displayed.

`\specialdiagnum`
- May be used to suppress the default diagram numbering (which uses a counter) and instead directly providing a diagram "number" which may be an arbritary text. This may also be used to suppress displaying a diagram number by providing an empty argument `{}`.

`\sourcenr`
- May be used to specify the number which was used for the problem inside an originals section.

`\source`
- May be used to specify the book or magazine where the problem was issued first.

`\issue`
- May be used to specify e. g. the issue of a magazine where the problem was issued.

`\pages`
- May be used to specify the page (or pages) where the problem was issued.

`\day` `\month` `\months` `\year`
- May be used to specify the different parts of the date of publication of the problem. (E. g. for problems issued in the german magazine ***Die Schwalbe*** you will typically only specify the `\month` and the `\year`. For problems issued in ***feenschach*** you may specify a period of months like `\months{7-10}`.)

`\tournament` `\award`
- May be used to specify an award and a tournament for the problem.

`\dedication` `\dedic`
- May be used to specify a dedication which was given by the author of the problem.

`\condition` `\cond`
- May be used to specify the fairy conditions of a problem. Different conditions should be separated with "; ".

`\twins`
- May be used to specify the different twins of a problem. Different twins should be separated with "; ".

`\remark` `\rem`
- May be used to specify remarks to the problem. I typically use this to explain fairy pieces on the board. You may also use the abbreviation `\rem`.

`\piecedefs`
- May be used to explain rotated pieces. An example:
  `\piecedefs{{ws}{TL}{Turm-L\"aufer-J\"ager}; {wn}{SU}{Nachtreiter}}`
  will create
   = Turm-Läufer-Jäger
   = Nachtreiter
  under the diagram.

`\solution` `\sol`
- `\solution` may be used to specify the solution of the problem. Normally this information is not used while displaying the board but it is only collected and may be put into your text using `\putsol`. There is also an abbreviation `\sol`.

`\judgement`
- May be used to describe the judgement given for a problem, e. g. when you are working on an award or when you are selecting problems for a "best of . . . " book.

`\comment`
- May be used to specify some comment on the problem (e. g. the authors original comment.)

`\themes`
- May be used to specify themes displayed in the problem. Different themes should be separated with ", ". When creating a theme index, the themes will automatically be used to create the register.

`\genre`
- May be used to specify genre of the problem. Different genres should be separated with ", ". The values are intended to using LaTeXimport within the PDB.

When providing an empty argument to commands `\award`, `\after`, `\dedic`, `\correction` and `\version` only a warning is issued to the logfile. In previous versions of `diagram.sty` using empty arguments with the mentioned commands produced empty lines above the diagram.

There are some commands which not only collect information but normally direct result in a change of the diagram. These are:

`\verticalcylinder`
- does not display the outer vertical lines to symbolize a verticalcylindric board.

`\horizontalcylinder`
- does not display the outer horizontal lines to symbolize a horizontalcylindric board.

`\noframe`
- does completely suppress the outer frame e. g. to symbolize a torus board.

`\noinnerframe`
- sometimes you need to suppress the inner frame instead of the outer frame which is achieved by using `\noinnerframe`. You may not use this together with `\noframe`.

`\gridchess`
- displays lines to seperates fieldsections for gridchess.

### 1.2.2 Modifying the layout of the diagram (and the solution)

There are a couple of switches which control the layout of the diagrams. These are typically used more generally, so you may specify these switches outside the `diagram` environment or use them in your own style, which depends on `cpd.sty`.

There are some switches which control the layout of the information which is displayed above a diagram:

`\diagleft`
- displayes the information left aligned

`\diagcenter`
- displayes the information centered

`\diagright`
- displayes the information right aligned

`\widedias`
- is like `\diagcenter` but the information shown above the diagram may span the whole width of the page. So LaTeX will not wrap long author names.

`\dianamestyle` `\solnamestyle`
Using `\dianamestyle` (or `\solnamestyle`) you may specify how authornames are written above the boards (or before the solutions). You may use this only if you use `\reversednames` (which is the default). Otherwise it is not possible to distuingish between firstname and sirname. You must specify one of the following options as parameter to `\dianamestyle` (or `\solnamestyle`):

**fullname** Writes the authorname as *firstname sirname*. This is the default.

**sirname** Writes the *sirname* only.

**short** Writes an abbreviation of the *firstname* and the *sirname*. The abbreviation is calculated as follows:

- The first letter of the *firstname* will be used.
  `\author{Brand, Thomas}` will be displayed as **T. Brand**
- When there is a combined *firstname* separated with a hyphen, each first letter will be used. (see below)
  `\author{Reich, Hans-Peter}` will be displayed as **H.-P. Reich**
- When specifying the author name, you may provide the abbreviation for the firstname using the form *sirname, firstname/abbreviation.*
  `\author{Brand, Thomas/Th.}` will be displayed as **Th. Brand**

**noname** displayes nothing

`\diagnumbering`
The same way you may specify `\pagenumbering` you may specify the format the diagrams are numbered using `\diagnumbering` and `\pagenumbering` you may specify `arabic, Roman, roman, Alph or alph`. The default used is `arabic`. This command also switches the display for diagram numbers on.

`\setmonthstyle`
You may also specify the way a month is displayed using `\setmonthstyle`.

There are some boolean switches, which control whether a specific information is displayed. These are as follows:

`piececounter`
- This is a LaTeX boolean, which is used to specify whether the number of pieces is displayed below the board. So you may change its value using `\setboolean{piececounter}{true}` or `\setboolean{piececounter}{false}`.

showcomputer
\nocomputer
\showcomputer

- There is a boolean value `computer`, which controls whether the infomration about a computer proof is displayed or not. This value may be changed using `\setboolean{showcomputer}{true}` or `\setboolean{showcomputer}{false}` For backwards compatiblity we support the macros `\nocomputer` and `\showcomputer`.

showcity

- This is a boolean switch, which controls whether the informated gathered using the `\city` command is displayed. The default of this value is `false`.

showacademictitle

- This is a boolean switch, which controls whether academic titles `\Dr`, `\Prof` or `\ProfDr` — typically used within the `\author` command — are displayed. The default is `true`.

legend

- This boolean controls whether a legend is displayed. The default value of this value is `false`. When legends are displayed the distance between inner and outer frame is automatically adjusted.

\notcomputerproofedsymbol
\computerproofedsymbol

You may specify the text, which is used indicate, whether a problem is proofed by a computer. To specifiy the symbol for a problem, which is proofed, is created by `\computerproofedsymbol`. To specifiy the symbol for a problem, which is not computer proofed, is created by `\notcomputerproofedsymbol`. You may redefine these commands by standard LaTeX means (`\renewcommand`).

\selectelchfont

You may specify which font is used for the chesspieces. There are two possible fonts:

**pk** for the font which was originally used in the german magazine ***Problemkiste*** ♔♕♖♗♘♙♚♛♜♝♞♟

**fs** for the font which was first used (and was created for) the magazine ***feenschach*** ♔♕♖♗♘♙♚♛♜♝♞♟

\diagramx
\diagramxi
\diagramxii

In analogy to the defaults for fontsizes of a document you may specify sizes of the fonts used in a diagram. The default will be set according to the fontsize specified as the `\documentclass` option.

### 1.2.3 Other commands

\label

- This overrides the normal `\label` definition such that the diagram number is displayed when using `\ref` instead of the page number.

\diagnum

- This macro expects a number as a parameter. The number will be used to (re-)initialize the diagram number counter. With this command the output of diagram numbers also is switched on. It must be used outside the `diagram` environment. As an optional parameter you may specify something, which will be used as prefix before the automatically updated diagram numbers. E. g. the command `\diagnum[T-]{4}` will produce the following diagram numbers for the following diagrams: **T-4**, **T-5**, **T-6**, ...

## 1.3 Special boards

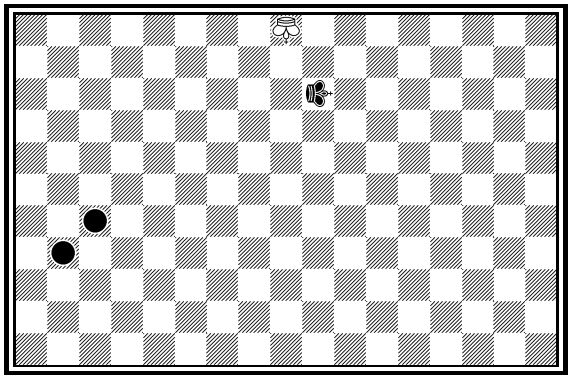
### 1.3.1 Changing the boardsize

diagram[]

Instead of using a boardsize of $8 \times 8$ some fairy problems need smaller or larger

boards. This can be achieved by specifying the rows and columns as an optional parameter to the `\begin{diagram}` environment. You first have to specify the columns and then the rows as the following examples shows.

**3**



C- (1+1)

is created by

```
\begin{diagram}[17x11]
\label{bigdia}
\pieces{wKUi{11}, sKRj9, sCc5b4}
\end{diagram}
```

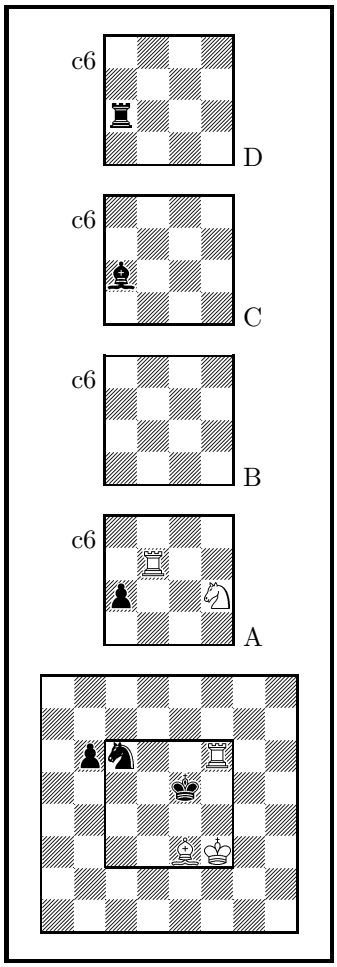
As you can see in the example, pieces are set using the `\pieces` macro. When using boards with more than 8 lines you have to continue with charachters **i, j, k, ...** In a board with more than 9 rows you have to specify the rows in curly braces { } as shown in the example.

### 1.3.2 Stereo- and Space-Chess-Diagrams

`stereodiagram`
`spacediagram[]`

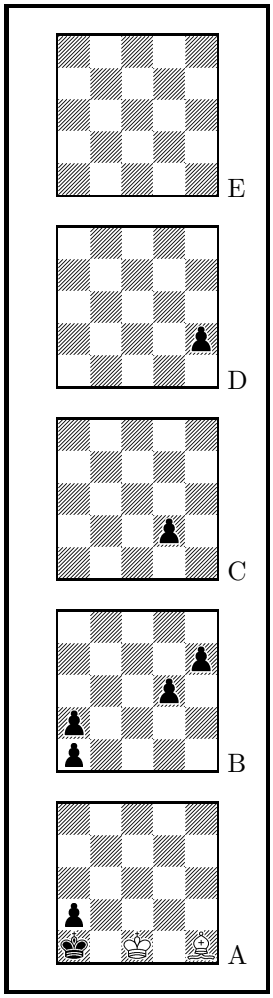
Other boards which are used from time to time are stereochess or spacechess boards (although there are quite few people which really have such boards!). To create these boards you just have to use either the `stereodiagram` or `spacediagram` environment instead of the normal `diagram` environment. Here is an example:

**4**
**Gerhard W. Jensch**
***3104. feenschach 1980***
*Preis*




#9 C- (5+6)

**5**
**T. R. Dawson**
***6595. Fairy Chess Review 12/1945***




#2 C- (2+8)

These diagrams have been produced by the following code:

```
\begin{stereodiagram}
\author{Jensch, Gerhard W.}
\sourcenr{3104.}
\source{feenschach}
\year{1980}
\award{Preis}
\pieces{wKf3, wTf6d5A, wLe3, wSf4A, sKe5, sTc4D, sLc4C, sSc6, sBb6c4A}
\stip{\#9}
\end{stereodiagram}
\hfill
\begin{spacediagram}
\author{Dawson, T. R.}
\sourcenr{6595}.
```

```
\source{Fairy Chess Review}
\month{12}
\year{1945}
\pieces{wKc1A, wLe1A, sKa1A, sBa2Aa1Ba2Bd3Be4Bd2Ce2D}
\stip{\#2}
\end{spacediagram}
```

The main change is within the notation of the pieces, but people knowing space- or stereo-chess problems see that the notation is just one would expect.

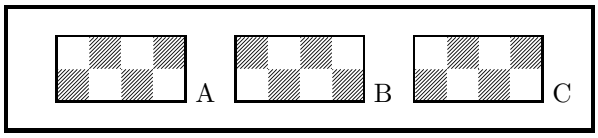
`\spacelayout` Sometimes one would like show the different planes of a space diagram from left to right. This may be switched using the `\spacelayout` command, which takes one parameter:

**vertical** for planes organized bottom up

**horizontal** for planes organized left to right

Is produced by

**6**




C- (0+0)

```
\begin{spacediagram}[4x2x3]
\spacelayout{horizontal}
\end{spacediagram}
```

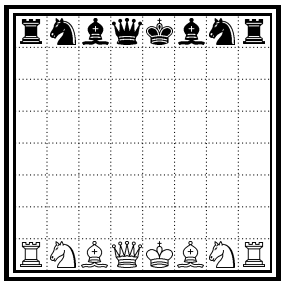
### 1.3.3 Cylindric boards / suppressing frames

`\horizontalcylinder` `\verticalcylinder` `\noframe` `\noinnerframe` To stylize a cylindric board one typically does not show parts of the frame. When using `\verticalcylinder` the horizontal lines of the outer frame will not be drawn. `\horizontalcylinder` suppresses the drawing of the vertical lines of the outer frame. Using `\noframe` completely supresses the outer frame. `\noinnerframe` suppresses the innerframe. In case of stereo- or space-chess-diagrams `\verticalcylinder`, `\horizontalcylinder` and `\noframe` suppresses the inner frame.

## 1.4 Change the coloring of the fields

`\allwhite` The `allwhite` boolean can be used to have all white squares. Therefore dotted lines are produced to separate the squares. For convenience we provide a command `allwhite` which switches the value of the `allwhite` boolean to true.

**7**

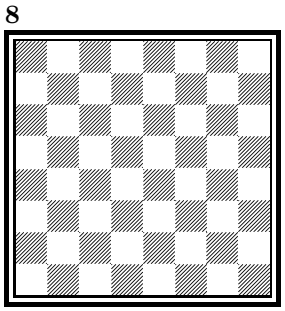


C- (8+8)

This was produced by:

```
\begin{diagram}
\allwhite
\pieces{wKe1, wDd1, wTa1h1, wLf1c1, wSb1g1, %
    sKe8, sDd8, sTa8h8, sLf8c8, sSb8g8}
\end{diagram}
```

\switchcolors The boolean `switchcolors` may be used to switch the coloring of white and black fields. For convenience we provide a command `switchcolors` which switches the value of the `switchcolors` boolean to true.




C- (0+0)

### 1.4.1 figurine Notation

figurine Instead of using the `diagram`, `stereodiagram` or `spacediagram` environment one may use the `figurine` environment. This suppresses the diagram output and produces a figurine notation inside the current text.

### 1.4.2 Changes within the board

\nofields You may remove single fields by using the `\nofields` or `\nosquares` command.
\nosquares Using this command does make sense for empty black fields only. This command expects a list of squares separated by ", ". You may also use this command within a stereo- or space-diagram. In this case you must specify the fields the same way you do it inside the `\pieces` command.

\fieldframe You may specify single fields, which should be surrounded by a frame. This is possible using the `\fieldframe` command. You must specify the list of fields which should have frames the same way you specify fields within the `\nofields` command.

\gridlines A more general form of lines within diagrams is possible by using the `\gridlines` command. You may specify a list of horizontal or vertical lines within the diagram. Different lines should be separated by ", ". A single line must be specified as:

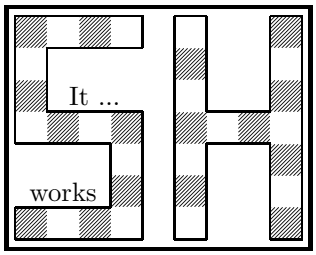
[plane](v or h)(x-coordinate)(y-coordinate)(length in squares)

You must specify a plane in case of stereo- or space-chess only. For a vertical line starting at the lower left corner of "c2" ending at the upper left corner of "c8" the command to use is: `\gridlines{v217}`. Concerning the coordinates and length specifications you should pay attention to put values greater 9 in curly braces { }.

\fieldtext Sometimes you need to show text on some squares. This is done using the `\fieldtext` command. The syntax for a single text is: {Text}(x-coordinate)(y-coordinate)

Now an example how to use `\gridlines`, `\nofields` and `\fieldtext` to create some "*Letter-Board*" with text inside.

**9**




C- (0+0)

```
\begin{diagram}[9x7]
\noinnerframe
\nofields{a2, b2, c2, a3, b3, c3, %
 b5, c5, d5, b6, c6, d6, %
 e1, e2, e3, e4, e5, e6, e7, %
 g1, h1, h2, h2, g3, h3, g5, h5, g6, h6, g7, h7}
\gridlines{h004, h013, h033, h143, h163, h074, %
 v001, v034, v142, v312, v404, v461, %
 h501, h571, h632, h642, h801, h871, %
 v507, v603, v643, v803, v843, v907}
\fieldtext{{It ...}c5, {works}b2}
\end{diagram}
```

## 1.5 Misc

### 1.5.1 Chess pieces within normal text

Sometimes you may need symbols of chess pieces within your normal text, e. g. to show the *Viele-Väter-Stellung* ♔c8, ♙b6, ♚a8, ♟a7. This is possible by `{\wK}c8, {\wB}b6, {\sK}a8, {\sB}a7`. Additionally you may use some of these symbols:

`\swL` a white bishop on a black square

`\ssL` a black bishop on a black square

`\wNr` a white nightrider

`\nNr` a neutral nightrider

`\sNr` a black nightrider

`\wGh` a white grashopper

`\nGh` a neutral grashopper

`\sGh` a black grashopper

`\Imi` ● an imitator, you may also use the **Circle** notation:

`\wC` ○ a white circle

`\nC` ◐ a neutral circle

`\sC` ● a black circle

`\wE` ⋈ a white equihopper

`\sE` ⋈ a black equihopper

`\nE` ⋈ a neutral equihopper

`\wX` ⧖ a white rotated equihopper

`\sX` ⧗ a black rotated equihopper

`\nX` ⧗ a neutral rotated equihopper

### 1.5.2 Other often used symbols

The style also defines commands for other symbols, which are often used within the declaration of twins or when writing a solution:

`\set` * setplay

`\ra` → a left to right arrow

`\lra` ↔ a double ended arrow

`\OO` **0-0** king side castling

`\OOO` **0-0-0** queen side castling

`\x` × for "takes"

`\any` ∼ for an arbitrary move (you must not simply use a ~ within your text because TEX handles this as a protected space)

`arrowskip` It is possible, to define a prefix and/or a suffix to use before and after arrows - typically a common separating space. Per default there is no such space. You may use the `arrowskip` environment have a common prefix and suffix before and after arrows:

```
\begin{arrowskip}{a\,}{\,b}
...
\end{arrowskip}
```

... will prefix arrows produced with `\ra` or `\lra` with `a\,` and suffix with `\,b`.

### 1.5.3 Internationalization

`\DefinePieces` This part is relevant for people who do not like the german notation for pieces and therefore want to change this within their sources. Using the german notation, you specify the color of a piece as **w**, **s** or **n**, the type of a piece as **K**, **D**, **T**, **L**, **S**, **B** and a possible rotation of a piece as **L**, **R** or **U**. To use another notation you may use the `\DefinePieces` command which takes 3 parameters.

1. the letters used to specify the colors of the pieces using the order white, black, neutral
2. the letters used to specify the type of a piece using the order king, queen, rook, bishop, knight, pawn. You may not use the characters **C, E** and **X**, because these are used for Circle, Equihopper and rotated Equihopper.
3. the letters used to specify an optional rotation using the order left-turned, right-turned, upside-down. You must use capital letters for this.

When using a `\DefinePieces` command, the commands are changed to its next usage (or to the end of the document). The command not only changes the pieces you may use within the `\pieces` command but also defines commands to be used within normal text, as the following example shows:
`\DefinePieces{wbn}{KQRBNP}{LRU}`
`\wDU\bKR\bwB`
creates 

### 1.5.4 When writing books

\develop To simplify your writings you may use the macro \develop. This will create the following additional information during developement:

- when you use \label in your diagrams the label will be shown at the left upper corner of the diagram.
- The given label will also be shown inside the solution and also in any register entry.
- when you have specified a \judgement this information will be put into the solution.

Most books on chessproblems contain registers for authors, sometimes also on themes and sources. As you already collect all these information very detailed within the diagram environment the generation of registers is very simple.

\makeaindex \authorindex To create a registers of authors you need to put the \makeaindex command inside the preamble of your document. This instructs latex to write an intermediate file containing information about authors and the numbers of the diagrams.[1] After a first LaTeX run on your document, you need to convert the intermediate file. This may be done with the makeindex program, which will typically called like

```
makeindex -o <filename>.and <filename>.adx
```

The resulting register may be put into your document using the \authorindex command.

\makesindex \sourceindex \maketindex \themeindex Like an index for authors you may also create indices for sources and/or themes. For an source register you need to put \makesindex into your document preamble; for a theme register the command is \maketindex. The conversion commands for the intermediate files are

```
makeindex -o <filename>.snd <filename>.sdx
```

for the source register and

```
makeindex -o <filename>.tnd <filename>.tdx
```

for the theme register.

The source register is inserted into the text using \sourceindex and the theme register using \themeindex.

### 1.5.5 Other useful stuff

\solpar In some environments — like window — the use of \par leads to unwanted effects. Therefore we use the command \solpar inside the definition of \@dia@solution, which is used to display a single solution when using \putsol. You may use \renewcommand{\solpar} to provide another definition of \solpar in such situations.

\insidediagram The problem information collected by some commands may be used in different places (author names will be shown above the diagram and at the beginning of solutions when output using \putsol). Therefore we need some special handling of e.g. footnotes, to avoid creating the footnotemarks multiple times. Such

[1] Normally registers contain page numbers but with chess problems normally people refer to the diagram numbers.

commands should be used within `\insidediagram` as shown by the following example:

```
\begin{diagram}
\author{Else\insidediagram{\footnotemark}, Someone}
\end{diagram}
```

# 2 The documentation driver

The following code will generate the documentation. Since it is the first piece of code in the file, the documentation can be obtained by simply processing the file with LaTeX $2_\varepsilon$.

```
⟨*driver⟩
\documentclass[a4paper]{article}
\usepackage{doc}
\usepackage{diagram}
\EnableCrossrefs
\CodelineIndex
\RecordChanges
\begin{document}
\DocInput{diagram.dtx}
\end{document}
⟨/driver⟩
```

# 3 The implementation of the style

Specifies the preamble of our style file.

```
⟨*style⟩
\ProvidesPackage{diagram}[2025/02/21-tikz]
```

`\DefaultDiagramSize` The `\DefaultDiagramSize` may be used in code to switch to the default diagram size. As this depends on the documents default font size we use the same option and execute *10pt* as default.

```
\newcommand*{\DefaultDiagramSize}{}
\DeclareOption{10pt}{\renewcommand*{\DefaultDiagramSize}{\diagramx}}
\DeclareOption{11pt}{\renewcommand*{\DefaultDiagramSize}{\diagramxi}}
\DeclareOption{12pt}{\renewcommand*{\DefaultDiagramSize}{\diagramxii}}
\ExecuteOptions{10pt}

\ProcessOptions
\AtBeginDocument{\DefaultDiagramSize}

\RequirePackage{ifthen}
\RequirePackage{calc}
\RequirePackage{tikz}
\RequirePackage{xstring}
```

Now we declare some constants to unify its usage within the style file.

```
\chardef\f@ur=4
\chardef\@ight=8
\newcount\elchfont
```

```
\chardef\@pkelch=0
\chardef\@fselch=1

\newcount\dia@type

\newboolean{@textproblem}
\setboolean{@textproblem}{false}
\def\textproblem{\setboolean{@textproblem}{true}\let\@dia@stipulation=\relax}

\newboolean{@solafterdiagram}
\setboolean{@solafterdiagram}{false}
\def\solafterdiagram{\setboolean{@solafterdiagram}{true}\ignorespaces}

\newif\if@vframe\@vframetrue
\newif\if@hframe\@hframetrue
\newif\if@leaveOuter\@leaveOutertrue

\newif\if@shortform

\newif\ifspace@vertical
\def\spacehorizontal{\space@verticalfalse}

% \newif\ifdi@no
\newboolean{@cpd@numbering@global}
\newboolean{@cpd@numbering@local}
\newcounter{board@nr}
\renewcommand{\theboard@nr}{\thediag}
% \newif\iffigcnt
\newboolean{piececounter}

\newcounter{cpd@piece@index}

\newboolean{showcity}
\setboolean{showcity}{false}
\newboolean{showacademictitle}
\setboolean{showacademictitle}{true}
\newboolean{legend}
\setboolean{legend}{false}

\newcount\@blackfield
\newboolean{allwhite}
\setboolean{allwhite}{false}
\newcommand{\allwhite}{\setboolean{allwhite}{true}}
\newcounter{field@border}
\newcount\@whitefield
\newboolean{switchcolors}
\setboolean{switchcolors}{false}
\newcommand{\switchcolors}{\setboolean{switchcolors}{true}}
```

We have counters for each color to count the pieces on the board.

```
\newboolean{cpd@checkPieceCounts}
\newcounter{cpd@defWhitePieces}
\newcounter{cpd@defBlackPieces}
\newcounter{cpd@defNeutralPieces}
```

```
\newcounter{cpd@whitePieces}
\newcounter{cpd@blackPieces}
\newcounter{cpd@neutralPieces}

\newcommand{\cpd@stepcounterWhite}{\stepcounter{cpd@whitePieces}}
\newcommand{\cpd@stepcounterBlack}{\stepcounter{cpd@blackPieces}}
\newcommand{\cpd@stepcounterNeutral}{\stepcounter{cpd@neutralPieces}}
\global\let\cpd@stepcounterPieces\relax

\newcount\help@a

\newbox\dia@box
\newbox\@cnt@box
\newdimen\@cnt@wd
\newbox\@stip@box

\newdimen\topdist\topdist\z@
\newbox\@test@box
\newdimen\@test@dimen
\newif\if@left

\newdimen\dia@lineskip

\newdimen\board@width
\newdimen\bd@width
\newdimen\head@width
\newdimen\sq@width

\newdimen\grid@width
\newdimen\inner@frame
\newdimen\outer@frame
\newdimen\space@frame
\newdimen\v@frame@dist
\newdimen\h@frame@dist
\newdimen\space@frame@dist
\newdimen\v@space@dist
\newdimen\h@space@dist

\newbox\sq@box
\newbox\plane@box
```

We need a lot of token registers to register the information from within the `diagram` environment. These token registers are defined here. Initially each token register is defined to contain `\relax`, which serves as an *end-marker* when parsing lists.

```
\newtoks\typis@tk\typis@tk={\relax}
\newtoks\label@tk\label@tk={\relax}
\newtoks\sol@tk\sol@tk={\relax}
\newtoks\number@tk\number@tk={\relax}
\newtoks\aut@tk\aut@tk={\relax}
\newtoks\city@tk\city@tk={\relax}
\newtoks\sourcenr@tk\sourcenr@tk={\relax}
\newtoks\source@tk\source@tk={\relax}
```

```
\newtoks\day@tk\day@tk={\relax}
\newcount\from@month\from@month=\z@
\newcount\to@month\to@month=\z@
\newtoks\year@tk\year@tk={\relax}
\newtoks\issue@tk\issue@tk={\relax}
\newtoks\pages@tk\pages@tk={\relax}
\newtoks\tournament@tk\tournament@tk={\relax}
\newtoks\award@tk\award@tk={\relax}
\newtoks\after@tk\after@tk={\relax}
\newtoks\version@tk\version@tk={\relax}
\newtoks\correction@tk\correction@tk={\relax}
\newtoks\dedic@tk\dedic@tk={\relax}
\newtoks\fidealbum@tk\fidealbum@tk={\relax}
\newtoks\theme@tk\theme@tk={\relax}
\newtoks\twins@tk\twins@tk={\relax}
\newtoks\judgement@tk\judgement@tk={\relax}
\newtoks\comment@tk\comment@tk={\relax}
\newtoks\computer@tk\computer@tk={-}
\newtoks\nofields@tk\nofields@tk={\relax}
\newtoks\fieldframe@tk\fieldframe@tk={\relax}
\newtoks\gridlines@tk\gridlines@tk={\relax}
\newtoks\pieces@tk\pieces@tk={\relax}
\newtoks\fen@tk\fen@tk={\relax}
\newtoks\fieldtext@tk\fieldtext@tk={\relax}
\newtoks\text@tk\text@tk={\relax}
\newtoks\stipulation@tk\stipulation@tk={\relax}
\newtoks\condition@tk\condition@tk={\relax}
\newtoks\remark@tk\remark@tk={\relax}
\newtoks\piecedefs@tk\piecedefs@tk={\relax}
\newtoks\@cpd@emptytest
```

To remember, which information has been specified, we define TEX-booleans for each command.

```
\newif\if@label\@labelfalse
\newif\if@number\@numberfalse
\newif\if@special\@specialfalse
\newif\ifauth@r\auth@rfalse
\newif\if@city\@cityfalse
\newif\if@sourcenr\@sourcenrfalse
\newif\if@source\@sourcefalse
\newif\if@date\@datefalse
\newif\if@day\@dayfalse
\newif\if@year\@yearfalse
\newif\if@issue\@issuefalse
\newif\if@pages\@pagesfalse
\newif\if@tournament\@tournamentfalse
\newif\if@award\@awardfalse
\newif\if@after\@afterfalse
\newif\if@version\@versionfalse
\newif\if@correction\@correctionfalse
\newif\if@dedication\@dedicationfalse
\newif\if@fidealbum\@fidealbumfalse
\newif\if@twins\@twinsfalse
\newif\if@theme\@themefalse
```

```
\newif\if@computer\@computerfalse
\newif\if@judgement\@judgementfalse
\newif\if@comment\@commentfalse
\newif\if@pieces\@piecesfalse
\newboolean{@cpd@fen}\setboolean{@cpd@fen}{false}%
\newif\if@fieldtext\@fieldtextfalse
\newif\if@nofields\@nofieldsfalse
\newif\if@gridlines\@gridlinesfalse
\newif\if@fieldframe\@fieldframefalse
\newif\if@stdgrid\@stdgridfalse
\newboolean{showcomputer}\setboolean{showcomputer}{true}%
\newcommand*{\computerproofedsymbol}{C+}
\newcommand*{\notcomputerproofedsymbol}{C-}
% \newif\if@show@computer\@show@computertrue
\newif\if@stipulation\@stipulationfalse
\newif\if@condition\@conditionfalse
\newif\if@remark\@remarkfalse
\newif\if@piecedefs\@piecedefsfalse
\newif\if@typis\@typisfalse
\newif\if@widedias\@widediasfalse
\newif\ifx@twins\x@twinsfalse
\newif\ifx@cond\x@condfalse
\newif\ifimitator\imitatorfalse
\newif\ifnormal@names\normal@namesfalse
\newif\ifs@lu
\newif\if@develop\@developfalse
\newif\if@notfirst
\newif\if@first
\newwrite\s@lfd
\let\below@newline=\relax
% These are used by the ``old'' board creating mechanism
```

The following counters are used when creating the diagram itself.

```
\newcounter{cpd@planes}
\newcounter{cpd@rows}
\newcounter{cpd@lines}
\newcounter{cpd@plane}
\newcounter{cpd@current@plane}
\newcounter{cpd@current@plane@display}
\newcounter{cpd@row}
\newcounter{cpd@line}
\newcounter{cpd@row@display}
\newcounter{cpd@line@display}
\newcounter{cpd@maxsquare}
\newcounter{cpd@helper}
\newcounter{cpd@current@square@index}
\newcounter{cpd@current@square@value}
```

Some boolean TEX-switches used whithin stereo- or spacechess diagrams.

```
\newif\if@stereo\@stereofalse
\newif\if@space\@spacefalse
```

These boolean switches are used to control the output of registers.

```
\newif\if@aindex\@aindexfalse
\newif\if@sindex\@sindexfalse
```

```
\newif\if@tindex\@tindexfalse
\newif\ifds@label
```

\cpd@begindiagram@hook
\cpd@enddiagram@hook

We define hooks to be executed in `\begin{diagram}` and `\end{diagram}`.

```
\newcommand{\cpd@begindiagram@hook}{}
\newcommand{\cpd@enddiagram@hook}{}
```

\diagram
\@diagram

Defines the code executed in `\begin{diagram}`. In case no optional size is given, a normal 8×8 board is generated.

```
\def\diagram{%
    \begingroup%
    \@ifnextchar [{\@diagram}{\@diagram[\@ight x\@ight]}%
}

\def\@cpd@initsize#1#2{%
    \setcounter{cpd@lines}{#1}%
    \setcounter{cpd@rows}{#2}%
    \setcounter{cpd@maxsquare}{\value{cpd@rows}*\value{cpd@lines}}%
}

\def\@diagram[#1x#2]{%
    \@cpd@initsize{#1}{#2}%
    \setcounter{cpd@plane}{0}%
    \setcounter{cpd@current@plane}{0}%
    \let\put@sqs=\put@sqs@normal%
    \let\read@plane=\read@plane@normal%
    \@start@diagram%
}

\def\stereodiagram{%
    \begingroup%
    \inner@frame=0.6pt%
    \@stereotrue%
    \@cpd@initsize{8}{8}%
    \let\put@sqs=\put@sqs@stereo%
    \let\read@plane=\read@plane@stereo%
    \@start@diagram%
}
\def\spacediagram{%
    \begingroup%
    \inner@frame=0.6pt%
    \@spacetrue%
    \@ifnextchar [{\@spacediagram}{\@spacediagram[5x5x5]}%
}

\def\@spacediagram[#1x#2x#3]{%
    \setcounter{cpd@planes}{#3}%
    \@cpd@initsize{#1}{#2}%
    \let\put@sqs=\put@sqs@space%
    \let\read@plane=\read@plane@space%
    \@start@diagram%
}
\def\@start@diagram{%
    \init@vars%
```

```
    \let\author=\ds@author%
    \let\day=\ds@day%
    \let\month=\ds@month%
    \let\year=\ds@year%
    \let\label=\ds@label%
    \cpd@begindiagram@hook%
    \ignorespaces%
}

\def\showtypis#1{%
    \@typistrue%
    \typis@tk={#1}%
    \ignorespaces%
}


\newboolean{@cpd@inside@diagram}
\setboolean{@cpd@inside@diagram}{false}
\newcommand{\insidediagram}[1]{%
     \ifthenelse{\boolean{@cpd@inside@diagram}}{#1}{}%
}
\def\enddiagram{%
    \let\author=\orig@author%
    \let\day=\orig@day%
    \let\month=\orig@month%
    \let\year=\orig@year%
    \let\label=\orig@label%
    \if@number%
    \else%
       \refstepcounter{board@nr}% so \label and \ref work properly
    \fi%
    %
    % Now \label@tk should be set, if wanted, so
    % we can generate the index entries
    %
    \@aindex%
    \@sindex%
    \@tindex%
    %
    % Now \@currentlabel will be set right, so we can use
    % the original label
    \if@label%
       \expandafter\@set@label\the\label@tk;%
    \fi%
    %
    % Now we know, if we have frames so we can setup our dimensions
    %
    \global\sq@width=\fontdimen\tw@\chessfont%
    \if@stereo%
       \bd@width=\@ight\sq@width%
       \board@width=\@ight\sq@width%
       \ifdim\h@frame@dist<\sq@width%
          \h@frame@dist=\sq@width%
       \fi%
```

```
332       % We do already skip with \v@space@dist
333       % So we use the additional skip \space@frame@dist here
334       \v@frame@dist=\space@frame@dist%
335       \ifdim\space@frame>\outer@frame%
336          \outer@frame=\space@frame%
337       \fi%
338       \advance\bd@width\tw@\inner@frame%
339       \advance\board@width\tw@\inner@frame%
340       \advance\board@width\tw@\h@frame@dist%
341       \advance\board@width\tw@\outer@frame%
342    \else\if@space%
343       \ifdim\h@frame@dist<1.5\sq@width%
344          \h@frame@dist=1.5\sq@width%
345       \fi%
346       % We do already skip with \v@space@dist
347       % So we use the additional skip \space@frame@dist here
348       \v@frame@dist=\space@frame@dist%
349       \ifdim\space@frame>\outer@frame%
350          \outer@frame=\space@frame%
351       \fi%
352       \ifspace@vertical%
353          \bd@width=\value{cpd@lines}\sq@width%
354          \board@width\bd@width%
355          \advance\bd@width\tw@\inner@frame%
356          \advance\board@width\tw@\inner@frame%
357          \advance\board@width\tw@\h@frame@dist%
358          \advance\board@width\tw@\outer@frame%
359       \else%
360          \bd@width=\value{cpd@lines}\sq@width%
361          \advance\bd@width\tw@\inner@frame%
362          \ifdim\h@space@dist<1.5\sq@width%
363             \h@space@dist=1.5\sq@width%
364          \fi%
365          %\h@space@dist=0.7\sq@width%
366          % Now we can compute the width of the complete board
367          \board@width\bd@width%
368          \advance\board@width\h@space@dist%
369          \multiply\board@width\value{cpd@planes}%
370          \advance\board@width\h@space@dist%
371          \advance\board@width\tw@\outer@frame%
372       \fi%
373    \else%
374       \ifthenelse{\boolean{legend}}{\v@frame@dist=1.5em\h@frame@dist=1.5em}{}%
375       \bd@width=\value{cpd@lines}\sq@width%
376       \ifthenelse{\value{cpd@lines} > 8}{%
377          % Make the board wider
378          \board@width=\value{cpd@lines}\sq@width%
379       }{%
380          % Make a normal width
381          \board@width=\@ight\sq@width%
382       }%
383       \advance\bd@width\tw@\inner@frame%
384       \advance\board@width\tw@\inner@frame%
385       \advance\board@width\tw@\h@frame@dist%
```

```
386       \advance\board@width\tw@\outer@frame%
387    \fi\fi%
388    \if@widedias%
389       \head@width=\textwidth%
390    \else%
391       \head@width=\board@width%
392    \fi%
393    %
394    % Now we should build the diagram itself
395    %
396    \ifthenelse{\boolean{@textproblem}}{%
397       % Put the stipulation into the \sq@box
398       \setbox\sq@box=\hbox{\vbox to \board@width{\hsize\board@width%
399          \stipfont%
400          \raggedright%
401          \sloppy%
402          \the\stipulation@tk%
403          \vfil%
404       }}%
405    }{%
406       \put@sqs% This builds up the \sq@box
407       % Check, if the given number of pieces is reached
408       \ifthenelse{\boolean{cpd@checkPieceCounts}}{%
409         \ifthenelse{\value{cpd@defWhitePieces}=\value{cpd@whitePieces}}{}%
410             {\errmessage{Wrong number of white pieces}}%
411         \ifthenelse{\value{cpd@defBlackPieces}=\value{cpd@blackPieces}}{}%
412             {\errmessage{Wrong number of black pieces}}%
413          \ifthenelse{\value{cpd@defNeutralPieces}=\value{cpd@neutralPieces}}{}%
414              {\errmessage{Wrong number of neutral pieces}}%
415       }{}%
416    }%
417    %
418    \global\setbox\dia@box=\hbox{\vbox{%
419       \setboolean{@cpd@inside@diagram}{true}%
420       \parindent\z@%
421       \parskip\z@%
422       \baselineskip11\p@\advance\baselineskip\dia@lineskip%
423       \hsize\head@width%
424       \centering%
425       % diagram header
426       \vskip\topdist%
427       \vbox{\hsize\board@width\hbox{%
428          \if@develop\if@label%
429             \noindent\raggedright\llap{\labelfont\the\label@tk\ }%
430          \fi\fi%
431          \vbox{%
432             \he@dpos\dia@above%
433          }%
434       }}%
435       \vskip\tw@\p@%
436       % diagram itself
437       \vtop{\hsize\board@width%
438          \hbox to \head@width{\hss\vbox{%
439             \hsize\board@width%
```

```
            \ifthenelse{\boolean{@textproblem}}{%
               \box\sq@box%
            }{%
               \outer@henbox{\box\sq@box}%
            }%
         }\hss}%
         % diagram trailer
         \hbox to \head@width{\hss\vtop{%
            \hsize\board@width%
            \parskip\z@%
            \raggedright%
            \put@count%
            \dia@below%
         }\hss}%
      }%
      \setboolean{@cpd@inside@diagram}{false}%
   }}% End of \dia@box
   \do@dia@job%
   \cpd@enddiagram@hook%
   \endgroup%
}

\def\do@put@count{%
   \ \ (\arabic{cpd@whitePieces}+\arabic{cpd@blackPieces}%
   \ifthenelse{\value{cpd@neutralPieces}>0}{+\arabic{cpd@neutralPieces}}{})%
}

\def\put@count{%
   % First we build the box with the figure count
   \ifthenelse{\boolean{showcomputer}\OR\boolean{piececounter}}{%
     \global\setbox\@cnt@box=\hbox{%
        \stipfont%
        \ifthenelse{\boolean{showcomputer}}{%
           \ \ \if@computer\computerproofedsymbol\else\notcomputerproofedsymbol\fi%
        }{}%
        \ifthenelse{\boolean{piececounter}}{%
           \do@put@count%
        }{}%
      }%
      \@cnt@wd=\wd\@cnt@box%
      \hangindent-\@cnt@wd%
      \hangafter\m@ne%
      \noindent%
      \hbox to \z@{%
        \hbox to \board@width{\hfil\unhbox\@cnt@box}\hskip -\board@width%
      }%
   }{}%
}

\let\endstereodiagram=\enddiagram
\let\endspacediagram=\enddiagram
\def\figurine{%
   \begingroup%
   \init@vars%
```

```
    \let\author=\ds@author%
    \let\day=\ds@day%
    \let\month=\ds@month%
    \let\year=\ds@year%
    \let\label=\ds@label%
    \cpd@begindiagram@hook%
}

\def\endfigurine{%
    \let\author=\orig@author%
    \let\day=\orig@day%
    \let\month=\orig@month%
    \let\year=\orig@year%
    \let\label=\orig@label%
    \if@number%
    \else%
       \refstepcounter{board@nr}% so \label and \ref work properly
    \fi%
    %
    % Now \label@tk should be set, if wanted, so
    % we can generate the index entries
    %
    \@aindex%
    \@sindex%
    \@tindex%
    %
    % Now \@currentlabel will be set right, so we can use
    % the original label
    %
    \if@label%
       \expandafter\@set@label\the\label@tk;%
    \fi%
    %
    \@show@figurine%
    \cpd@enddiagram@hook%
    \endgroup%
}
%
\gdef\selectelchfont#1{%
    \global\elchfont\csname @#1elch\endcsname\defaultelchfont%
}
```

Here we define commands to change fonts used for text above and below the diagram. You may redefine to adjust the fonts to your needs.

\authorfont
\cityfont
\sourcefont
\awardfont
\dedicfont
\stipfont
\remfont
\labelfont
\cpd@boardfont
\legendfont

```
\newcommand*{\authorfont}{\bfseries}
\newcommand*{\cityfont}{\slshape}
\newcommand*{\sourcefont}{\bfseries\itshape}
\newcommand*{\awardfont}{\itshape}
\newcommand*{\dedicfont}{\itshape}
\newcommand*{\stipfont}{\rmfamily}
\newcommand*{\remfont}{\rmfamily}
\newcommand*{\labelfont}{\rmfamily}
\newcommand*{\cpd@boardfont}{\rmfamily}
```

```
\newcommand*{\legendfont}{\sffamily}
```

We have three different default sizes for diagrams. The following commands switch fontsizes used for the chessfonts to typeset the diagrams.

\diagramx
\diagramxi
\diagramxii

```
\newcounter{cpd@chessfontsize}
\newcommand*{\diagramx}{
   \ifcase\elchfont\relax%
      \font\chessfont=pkelch12
      \font\chtextfont=pkelch10
   \else%
      \font\chessfont=fselch12
      \font\chtextfont=fselch10
   \fi%
   \dia@lineskip\z@
   \dia@type\z@
   \setcounter{cpd@chessfontsize}{12}%
}

\newcommand*{\diagramxi}{
   \ifcase\elchfont\relax%
      \font\chessfont=pkelch14
      \font\chtextfont=pkelch11
   \else%
      \font\chessfont=fselch14
      \font\chtextfont=fselch11
   \fi%
   \dia@lineskip\@ne\p@
   \dia@type\@ne
   \setcounter{cpd@chessfontsize}{14}%
}

\newcommand*{\diagramxii}{
   \ifcase\elchfont\relax%
      \font\chessfont=pkelch16
      \font\chtextfont=pkelch12
   \else%
      \font\chessfont=fselch16
      \font\chtextfont=fselch12
   \fi%
   \dia@lineskip\tw@\p@
   \dia@type\tw@
   \setcounter{cpd@chessfontsize}{16}%
}
```

\defaultelchfont

`\defaultelchfont` is used to define the fontsize used to typeset the diagrams depending on the documentsize.

```
\def\defaultelchfont{%
   \ifcase\@ptsize\relax%
      \diagramx\or%
      \diagramxi\or%
      \diagramxii%
   \fi%
}
```

```
\def\dianamestyle#1{\def\@dianame{\csname @#1\endcsname}}
\def\solnamestyle#1{\def\@solname{\csname @#1\endcsname}}
\newcommand*{\diagnum}[2][]{%
    \renewcommand*{\@dianumber@prefix}{#1}%
    \setcounter{board@nr}{#2}%
    \addtocounter{board@nr}{\m@ne}}
```

Now we define a couple of abbreviations and special symbols often used when setting problem chess documents.

\ra
\lra
\rla
arrowskip

Arrows and specification of space (or something different) before and after arrows.

```
\newcommand{\@cpd@pre@arrow}{}
\newcommand{\@cpd@post@arrow}{}
\newcommand{\ra}{\@cpd@pre@arrow\mbox{$\rightarrow$}\@cpd@post@arrow}
\newcommand{\lra}{\@cpd@pre@arrow\mbox{$\leftrightarrow$}\@cpd@post@arrow}
\let\rla=\lra
\newcommand*{\@cpd@prepost@arrow}[2]{%
    \renewcommand*{\@cpd@pre@arrow}{#1}
    \renewcommand*{\@cpd@post@arrow}{#2}
}
\newenvironment{arrowskip}{%
    \@cpd@prepost@arrow%
}{%
}
```

\x
\set
\OO
\OOO
\any
\further

```
\newcommand{\x}{\mbox{\ifmmode\times\else$\times$\fi}}
\def\set{\kern -.05em\raise .1ex\hbox{*}}
\def\@O{0\raise.25ex\hbox{-}\kern -.1em\relax}
\def\OO{\@OO}
\def\OOO{\@O\@OO}
\def\any{\ifmmode\sim\else$\sim$\fi}
\def\further{\ifmmode\Rightarrow\else$\Rightarrow$\fi\ \ignorespaces}

\def\spacelayout#1{\csname space@#1\endcsname}
\def\nodiagnumbering{%
    % \global\di@nofalse
    \setboolean{@cpd@numbering@global}{false}
}
\newcommand*{\@dianumber@prefix}{}
\def\diagnumbering#1{%
    \setboolean{@cpd@numbering@global}{true}%
    % \global\di@notrue%
    \diagnum{\@ne}%
    \gdef\thediag{\@dianumber@prefix\csname @#1\endcsname\c@board@nr}%
}
```

\diagcenter
\diagleft
\diagright

The macros \diagcenter, \diagleft and \diagright simply define the macro \he@dpos to the corresponding paragraph alignment.

```
\def\diagcenter{\def\he@dpos{\centering}}
\def\diagleft{\def\he@dpos{\raggedright}}
\def\diagright{\def\he@dpos{\raggedleft}}
```

\setmonthstyle The implementation of \setmonthstyle does \diagnumbering define a command which uses the given parameter as a part of the command name.

```
\def\setmonthstyle#1{\def\write@month{\csname @#1\endcsname}}

\def\specialdiagnum#1{%
    \ifthenelse{\equal{}{#1}}{%
        % We disable displaying the diagram number
        \setboolean{@cpd@numbering@local}{false}%
    }{%
        \setboolean{@cpd@numbering@local}{true}%
        \@specialtrue%
    }
    \number@tk={#1}\@numbertrue%
    \def\thediag{#1}\def\@currentlabel{#1}%
    \ignorespaces%
}
```

\ds@label \ds@author The macros \ds@label and \ds@author are defined internally and are made public within \begin{diagram}. This is because the macros \label and \author are normal LaTeX-macros and I want to avoid to redefine these globally.

```
\def\ds@label{%
    \@ifstar{\ds@labelfalse\ds@xlabel}{\ds@labeltrue\ds@xlabel}%
}
\def\ds@author#1{%
    \aut@tk={#1}\auth@rtrue%
    \ignorespaces%
}
```

\ds@academictitle \Dr \Prof \ProfDr

```
\def\ds@academictitle#1{\ifthenelse{\boolean{showacademictitle}}{#1~}{}\ignorespaces}
\newcommand{\Dr}{\ds@academictitle{Dr.}}
\newcommand{\Prof}{\ds@academictitle{Prof.}}
\newcommand{\ProfDr}{\ds@academictitle{Prof.\,Dr.}}

\def\@cpd@warnIfEmpty#1#2{%
    \begingroup%
        \@cpd@emptytest={#1}%
        \edef\@cpd@param{\the\@cpd@emptytest}%
    \expandafter\endgroup%
    \ifx\@cpd@param\@empty\relax%
        \message{^^JWARNING: empty '#2' argument.^^J}%
    \fi
}
\def\city#1{%
    \city@tk={#1}\@citytrue%
    \ignorespaces%
}
\def\sourcenr#1{%
    \sourcenr@tk={#1}\@sourcenrtrue%
    \ignorespaces%
}
\def\source#1{%
    \source@tk={#1}\@sourcetrue%
    \ignorespaces%
```

```
}
\def\ds@day#1{%
    \day@tk={#1}\@daytrue\@datetrue%
    \ignorespaces%
}
\def\ds@month#1{%
    \from@month=#1\@datetrue%
    \ignorespaces%
}
\def\months#1{%
    \@months#1;%
    \ignorespaces%
}
\def\ds@year#1{%
    \year@tk={#1}\@yeartrue\@datetrue%
    \ignorespaces%
}
\def\issue#1{%
    \issue@tk={#1}\@issuetrue%
    \ignorespaces%
}
\def\pages#1{%
    \pages@tk={#1}\@pagestrue%
    \ignorespaces%
}
\def\tournament#1{%
    \tournament@tk={#1}\@tournamenttrue%
    \ignorespaces%
}
\def\award#1{%
     \@cpd@warnIfEmpty{#1}{award}%
     \award@tk={#1}\@awardtrue%
     \ignorespaces%
}
\def\version#1{%
     \@cpd@warnIfEmpty{#1}{version}%
     \version@tk={#1}\@versiontrue%
     \ignorespaces%
}
\def\after#1{%
     \@cpd@warnIfEmpty{#1}{after}%
     \after@tk={#1}\@aftertrue%
     \ignorespaces%
}
\def\correction#1{%
     \@cpd@warnIfEmpty{#1}{correction}%
     \correction@tk={#1}\@correctiontrue%
     \ignorespaces%
}
\def\dedication#1{%
     \@cpd@warnIfEmpty{#1}{dedication}%
     \dedic@tk={#1}\@dedicationtrue%
     \ignorespaces%
}
```

```
\def\fidealbum#1{%
   \fidealbum@tk={#1}\@fidealbumtrue%
   \ignorespaces%
}
\def\pieces{%
   \@ifnextchar[%
   {\x@pieces}%
   {\@pieces}%
}
\def\x@pieces[#1]{%
   % We should parse the given piececounts
   \setboolean{cpd@checkPieceCounts}{true}%
   \@parseWhiteAndBlackCount#1+\e@list
   \@pieces%
}
\def\@parseWhiteAndBlackCount#1+#2+{%
   \setcounter{cpd@defWhitePieces}{#1}%
   \setcounter{cpd@defBlackPieces}{#2}%
   \futurelet\n@xt\cpd@checkNeutral%
}
\let\cpd@nextproc=\relax%
\def\cpd@checkNeutral{%
    \if\n@xt\relax%
        \let\cpd@nextproc=\relax%
    \else%
        \let\cpd@nextproc=\@parseNeutralCount%
    \fi%
    \cpd@nextproc%
}
\def\@parseNeutralCount#1+{%
   \setcounter{cpd@defNeutralPieces}{#1}%
}
\def\@pieces#1{%
   \pieces@tk={#1}\@piecestrue%
   \ignorespaces%
}
\newcommand{\fen}[2][]{%
   \ifthenelse{\equal{#1}{}}%
      {}% Do nothing
      {%
         \setboolean{cpd@checkPieceCounts}{true}%
         \@parseWhiteAndBlackCount#1+\e@list%
      }%
   \fen@tk={#2}\setboolean{@cpd@fen}{true}%
   \ignorespaces%
}
\def\fieldtext#1{%
   \fieldtext@tk={#1}\@fieldtexttrue%
   \ignorespaces%
}
\def\nofields#1{%
   \nofields@tk={#1}\@nofieldstrue%
   \ignorespaces%
}
```

```
\let\nosquares\nofields
\def\gridlines#1{%
    \gridlines@tk={#1}\@gridlinestrue%
    \ignorespaces%
}
\def\fieldframe#1{%
    \fieldframe@tk={#1}\@fieldframetrue%
    \ignorespaces%
}
\def\stipulation#1{%
    \stipulation@tk={#1}\@stipulationtrue%
    \ignorespaces%
}
\def\condition{%
    \@ifstar{\x@condtrue\@condition}{\@condition}%
}
\def\@condition#1{%
    \condition@tk={#1}\@conditiontrue%
    \ignorespaces%
}
\def\twins{%
    \@ifstar{\x@twinstrue\@twins}{\@twins}%
}
\def\@twins#1{%
    \twins@tk={#1}\@twinstrue%
    \ignorespaces%
}
\def\remark#1{%
    \remark@tk={#1}\@remarktrue%
    \ignorespaces%
}
\def\piecedefs#1{%
    \piecedefs@tk={#1}\@piecedefstrue%
    \ignorespaces%
}
% \def\@piecedef#1{\csname#1\x@piecedef\endcsname\l@@klist}
% \newcommand{\piecedef}[3][ws]{%
%     \def\x@piecedef{#2}%
%     \let\@action=\@piecedef%
%     \hbox{\l@@klist#1\e@list%
%     \ = #3}%
% }
\def\Co#1{%
    \ifx#1+\@computertrue\computer@tk={+}\fi%
    \ignorespaces%
}
\long\def\solution#1{%
    \sol@tk={#1}\global\s@lutrue%
    \ignorespaces%
}
\def\themes#1{%
    \theme@tk={#1}\@themetrue%
    \ignorespaces%
}
```

```
\def\genre#1{%
   \relax% Currently not used within diagram.sty
}
\long\def\comment#1{%
   \comment@tk={#1}\@commenttrue%
   \ignorespaces%
}
\long\def\judgement#1{%
   \judgement@tk={#1}\@judgementtrue%
   \ignorespaces%
}
\def\noframe{%
   \@vframefalse\@hframefalse%
   \ignorespaces%
}
\def\noinnerframe{%
   \@leaveOuterfalse\@vframefalse\@hframefalse%
   \ignorespaces%
}
\def\verticalcylinder{%
   \@vframefalse%
   \ignorespaces%
}
\def\horizontalcylinder{%
   \@hframefalse%
   \ignorespaces%
}
\def\stdgrid{%
   \@stdgridtrue%
   \ignorespaces%
}
```

\gridchess \magic \tourn \dedic \stip \cond \rem \sol

Here we define some abbreviations and synonyms for other macros.

```
\let\gridchess=\stdgrid
\let\magic=\fieldframe
\let\tourn=\tournament
\let\dedic=\dedication
\let\stip=\stipulation
\let\cond=\condition
\let\rem=\remark
\let\sol=\solution

\def\develop{%
   \@developtrue%
   \ignorespaces%
}
\def\showcomputer{%
   \setboolean{showcomputer}{true}%
   \ignorespaces%
}
\def\nocomputer{%
   \setboolean{showcomputer}{false}%
   \ignorespaces%
}
```

```
\def\putsol{\immediate\closeout\s@lfd\input\jobname.sol\cl@arsol}
\def\widedias{\@widediastrue\diagcenter}
\def\nowidedias{\@widediasfalse}
\def\normalnames{\normal@namestrue}
\def\reversednames{\normal@namesfalse}
\def\makeaindex{%
    \@dia@index%
    \newindex[thediag]{author}{adx}{and}{Autorenverzeichnis}%
    \@aindextrue\reversednames%
}

\def\makesindex{%
    \@dia@index%
    \newindex[thediag]{source}{sdx}{snd}{Quellenregister}%
    \@sindextrue%
}

\def\maketindex{%
    \@dia@index%
    \newindex[thediag]{theme}{tdx}{tnd}{Themenregister}%
    \@tindextrue%
}

\def\authorindex{{\let\@idxitem\@aidxitem\printindex[author]}}
\def\sourceindex{\printindex[source]}
\def\themeindex{\printindex[theme]}
\def\DefinePieces#1#2#3{%
    \@setPieceColor#1\@setPieceSpec#2\@setPieceRotation#3%
    \cpd@piecedef@rotations%
    \expandafter\xdef\csname\ds@black\ds@white\ds@bishop\endcsname{%
       \noexpand\ch@fig{20}%
    }%
    \expandafter\xdef\csname\ds@black\ds@black\ds@bishop\endcsname{%
       \noexpand\ch@fig{32}%
    }%
    \expandafter\xdef\csname\ds@white F\endcsname{{\chessfont\ }}
    \expandafter\xdef\csname\ds@black F\endcsname{{\chessfont\char144}}
    \expandafter\xdef\csname\ds@white Nr\endcsname{%
       \noexpand\ch@fig{109}%
    }%
    \expandafter\xdef\csname\ds@neutral Nr\endcsname{%
       \noexpand\ch@fig{115}%
    }%
    \expandafter\xdef\csname\ds@black Nr\endcsname{%
       \noexpand\ch@fig{121}%
    }%
    \expandafter\xdef\csname\ds@white Gh\endcsname{%
       \noexpand\ch@fig{112}%
    }%
    \expandafter\xdef\csname\ds@neutral Gh\endcsname{%
       \noexpand\ch@fig{118}%
    }%
    \expandafter\xdef\csname\ds@black Gh\endcsname{%
       \noexpand\ch@fig{124}%
```

```
    }%
    \expandafter\xdef\csname\ds@white C\endcsname{%
       \noexpand\ch@fig{145}%
    }%
    \expandafter\xdef\csname\ds@neutral C\endcsname{%
       \noexpand\ch@fig{151}%
    }%
    \expandafter\xdef\csname\ds@black C\endcsname{%
       \noexpand\ch@fig{157}%
    }%
}
\def\Imi{\ch@fig{157}}
\def\wE{\ch@fig{216}}
\def\nE{\ch@fig{222}}
\def\sE{\ch@fig{228}}
\def\wX{\ch@fig{180}}
\def\nX{\ch@fig{186}}
\def\sX{\ch@fig{192}}
\def\cpd@whiteField{{\chessfont\ }}
\def\cpd@blackField{{\chessfont\char144}}
```

`\dia@above` The content of the box above a diagram is controlled by the macro `\dia@above`. It just delegates the information to a couple of other macros, which then generate the displayed information above the diagram.

```
\newboolean{above@newline}
\newcommand{\above@newline}{\ifthenelse{\boolean{above@newline}}{\linebreak}{\setboolean{above@new
\def\dia@above{%
    \setboolean{above@newline}{false}%
    \@dia@number%
    \@dia@authors%
    \@dia@city%
    \@dia@after%
    \@dia@version%
    \@dia@source%
    \@dia@correction%
    \@dia@tournament%
    \@dia@award%
    \@dia@dedic%
    \@dia@fidealbum%
}
```

`\dia@below` As before, the macro `\dia@below` creates the displayed information below the chessboard - forwarding to a couple of other macros.

```
\def\dia@below{%
    \bgroup%
    \if@stipulation%
       \@dia@stipulation%
    \fi%
    \ifx@cond\else%
       \@dia@condition%
    \fi%
    \ifx@twins\else%
       \@dia@twins%
```

```
    \fi%
    \@dia@piecedefs%
    \@dia@remark%
    \ifthenelse{\boolean{@solafterdiagram}}{%
       \below@newline%
       \the\sol@tk%
    }{}%
    \noindent\hbox{}\newline\hbox{}%
    \egroup%
}
```

`\@dia@number` The `\@dia@number` macro simply creates the diagram number in a single paragraph.

```
\def\@dia@number{%
    %\ifdi@no\above@newline{\authorfont\thediag}\fi%
    \ifthenelse{\boolean{@cpd@numbering@local}}{%
       \above@newline{\authorfont\thediag}%
    }{}%
}
```

`\@dia@authors` This macro is used to create the list of authors specified within the `\author` macro inside the `diagram` environment. Depending on the TEX-boolean `normal@names` we either simply display the registered author or parse the list of authors by using the generic `\@parseTokenList` macro.

```
\def\@dia@authors{%
    \ifauth@r%
       \ifnormal@names%
          \above@newline
          {\authorfont\the\aut@tk}%
       \else%
          \let\@action=\@dia@writename% Parse the list of authors
        \@parseTokenlist\aut@tk;
       \fi%
    \fi%
}

\def\@show@city#1;{\if@notfirst\ \slash\ \else\@notfirsttrue\fi#1}

\def\p@rsecity#1; {\@show@city#1;\l@@klist}

\def\@dia@city{%
    \ifthenelse{\boolean{showcity}}{%
        \if@city%
           \above@newline%
           \bgroup%
           \cityfont\@notfirstfalse%
           \let\@action=\p@rsecity\@parseTokenlist\city@tk;%
           \egroup%
        \fi%
    }{}%
}

\def\@dia@after{%
    \if@after%
```

```
      \bgroup%
      \above@newline%
      \dedicfont\the\after@tk%
      \egroup%
   \fi%
}

\def\@dia@version{%
   \if@version%
      \above@newline%
      \bgroup%
      \dedicfont\the\version@tk%
      \egroup%
   \fi%
}

\def\@dia@date{%
   \ifnum\from@month>\z@%
      \if@day%
         \the\day@tk.\write@month\from@month%
      \else%
         \write@month\from@month%
      \fi%
      \ifnum\to@month>\z@--\write@month\to@month\fi%
      \if@day.\else/\fi%
   \fi%
   \if@year\the\year@tk\fi%
}

\def\@dia@source{%
   \if@source%
      \above@newline%
      \bgroup%
      \sourcefont%
      \if@sourcenr\the\sourcenr@tk\ \fi
      \the\source@tk%
      \if@date\ \ \fi\@dia@date%
      \if@issue\ \ \the\issue@tk\fi%
      \if@pages ,\ \the\pages@tk\fi%
      \egroup%
   \else%
      \if@tournament\else\if@date%
         \above@newline%
         \bgroup%
         \sourcefont%
         \@dia@date%
         \egroup%
      \fi\fi%
   \fi%
}

\def\@dia@correction{%
   \if@correction%
      \above@newline%
```

```
      \bgroup%
      \dedicfont\the\correction@tk%
      \egroup%
   \fi%
}

\def\@dia@tournament{%
   \if@tournament
      \above@newline%
      \bgroup%
      \awardfont%
      \the\tournament@tk
      \if@source\else\if@date%
         \ \ \@dia@date%
      \fi\fi%
      \egroup%
   \fi%
}

\def\@dia@award{%
   \if@award%
      \above@newline%
      \bgroup%
      \awardfont\the\award@tk%
      \egroup%
   \fi%
}

\def\@dia@dedic{%
   \if@dedication%
      \above@newline%
      \bgroup%
      \dedicfont\the\dedic@tk%
      \egroup%
   \fi%
}

\def\@show@album#1/#2;{#1 FIDE-Album #2}

\def\@dia@fidealbum{%
   \if@fidealbum%
      \above@newline%
      {\expandafter\@show@album\the\fidealbum@tk;}%
   \fi%
}

\def\@twinskip{\ \ }

\def\@dia@stipulation{%
   \if@stipulation%
      \bgroup%
      \stipfont%
      \the\stipulation@tk%
      \ifx@twins%
```

```
         \let\below@newline\@twinskip%
         \@dia@twins%
      \else\ifx@cond%
         \let\below@newline\@twinskip%
         \@dia@condition%
      \fi\fi%
      \egroup%
      \let\below@newline\newline%
   \else%
      \x@twinsfalse%
      \x@condfalse%
      \let\below@newline\relax%
   \fi%
}

\def\x@write@twin#1; {%
   \hskip1em#1%
   \@lefttrue\let\below@newline\newline%
   \let\@action\write@twins%
   \l@@klist%
}

\def\write@twins#1; {%
   \setbox\@test@box=\hbox{#1\if@left~~\fi}%
   \ifdim\wd\@test@box>4\sq@width%
      \below@newline%
      \@lefttrue%
      #1%
   \else%
      \if@left%
         \below@newline%
      \fi%
      \noindent\hbox to 4\sq@width{#1\hfil}%
      \if@left%
         \@leftfalse%
      \else%
         \@lefttrue%
      \fi%
   \fi%
   \let\below@newline\newline%
   \l@@klist%
}

\def\@dia@twins{%
   \if@twins%
      \bgroup%
      \@lefttrue%
      \remfont%
      \ifx@twins%
         \let\@action=\x@write@twin%
      \else%
         \let\@action=\write@twins%
      \fi%
      \@parseTokenlist\twins@tk;%
```

```
      \egroup%
      \let\below@newline\newline%
   \fi%
}

\def\@dia@condition{%
   \if@condition%
      \bgroup%
      \@lefttrue%
      \remfont%
      \ifx@cond%
         \let\@action=\x@write@twin%
      \else%
         \let\@action=\write@twins%
      \fi%
      \@parseTokenlist\condition@tk;%
      \egroup%
      \let\below@newline\newline%
   \fi%
}

\def\check@piecedef{%
    \ifx\next@piecedef\relax%
        \let\col@action=\relax%
    \else%
        \let\col@action=\@@piecedef%
    \fi%
    \col@action%
}
\def\@@piecedef#1{\csname#1\x@piecedef\endcsname\parse@piecedef}

\def\parse@piecedef{\futurelet\next@piecedef\check@piecedef}

\def\@piecedef#1#2#3{%
   \def\x@piecedef{#2}%
   \below@newline%
   \hbox{%
   \parse@piecedef#1\relax%
   \ = #3}%
}

\def\write@piecedefs#1; {%
   \@piecedef#1%
   \l@@klist%
}

\def\@dia@piecedefs{%
   \if@piecedefs%
      \bgroup%
      \@lefttrue%
      \let\below@newline\newline%
      \remfont\let\@action=\write@piecedefs%
      \@parseTokenlist\piecedefs@tk;%
      \egroup%
```

```
    \fi%
}

\def\@dia@remark{%
    \if@remark%
        \bgroup%
        \@lefttrue%
        \remfont\let\@action=\write@twins%
        \@parseTokenlist\remark@tk;%
        \egroup%
        \let\below@newline\newline%
    \fi%
}

\def\parse@params#1{%
    \ifcase\help@a\relax
        \label@tk={#1}\ifx\relax#1\else\@labeltrue\fi\or%
        \number@tk={#1}\ifx\relax#1\else\@numbertrue\fi\or%
        \aut@tk={#1}\ifx\relax#1\else\auth@rtrue\fi\or%
        \city@tk={#1}\ifx\relax#1\else\@citytrue\fi\or%
        \sourcenr@tk={#1}\ifx\relax#1\else\@sourcenrtrue\fi\or%
        \source@tk={#1}\ifx\relax#1\else\@sourcetrue\fi\or%
        \day@tk={#1}\ifx\relax#1\else\@daytrue\fi\or%
        \from@month=#1\or%
        \to@month=#1\or%
        \year@tk={#1}\ifx\relax#1\else\@yeartrue\fi\or%
        \issue@tk={#1}\ifx\relax#1\else\@issuetrue\fi\or%
        \pages@tk={#1}\ifx\relax#1\else\@pagestrue\fi\or%
        \tournament@tk={#1}\ifx\relax#1\else\@tournamenttrue\fi\or%
        \award@tk={#1}\ifx\relax#1\else\@awardtrue\fi\or%
        \after@tk={#1}\ifx\relax#1\else\@aftertrue\fi\or%
        \version@tk={#1}\ifx\relax#1\else\@versiontrue\fi\or%
        \correction@tk={#1}\ifx\relax#1\else\@correctiontrue\fi\or%
        \dedic@tk={#1}\ifx\relax#1\else\@dedicationtrue\fi\or%
        \theme@tk={#1}\ifx\relax#1\else\@themetrue\fi\or%
        \twins@tk={#1}\ifx\relax#1\else\@twinstrue\fi\or%
        \computer@tk={#1}\or%
        \comment@tk={#1}\ifx\relax#1\else\@commenttrue\fi\or%
        \judgement@tk={#1}\ifx\relax#1\else\@judgementtrue\fi\or%
        \sol@tk={#1}%
    \fi%
    \advance\help@a \@ne%
    \l@@klist%
}

\def\split@param#1{%
    \@labelfalse\@numberfalse\auth@rfalse\@cityfalse%
    \@sourcenrfalse\@sourcefalse\@dayfalse\@yearfalse%
    \@issuefalse\@pagesfalse\@tournamentfalse\@awardfalse%
    \@afterfalse\@versionfalse\@correctionfalse\@dedicationfalse%
    \@themefalse\@twinsfalse\@commentfalse\@judgementfalse%
    \help@a=\z@%
    \let\@action=\parse@params\l@@klist#1\e@list%
}
```

```
\newcommand{\solpar}{\par}
\def\@dia@solution{%
   \bgroup%
   \parindent\z@%
   \parskip\tw@\p@%
   {\bfseries%
      \noindent\if@label\showlabel{\the\label@tk}\fi%
      \if@number\the\number@tk) \fi%
      \ifauth@r%
         \ifnormal@names%
            \the\aut@tk%
         \else%
            {\@notfirstfalse% We are the first one
            \def\name@sep{, }%
            \let\@action=\@sol@writename%
            \@parseTokenlist\aut@tk;}:%
         \fi%
         \newline%
      \fi%
   }%
   \if@develop\if@judgement\the\judgement@tk\solpar\fi\fi%
   \the\sol@tk\solpar%
   \if@comment\the\comment@tk\solpar\fi%
   \egroup%
}
\grid@width=0.6\p@
\inner@frame=0.6\p@
\outer@frame=1.2\p@
\space@frame=\outer@frame
\v@frame@dist=\tw@\p@%
\h@frame@dist=\tw@\p@%
\space@frame@dist=\z@
\v@space@dist=1em
\def\@show@figurine{%
   \noindent%
   \@figurine@number%
   \@figurine@author%
   \@figurine@city%
   \@figurine@after%
   \@figurine@correction%
   \@figurine@version%
   \@figurine@source%
   \@figurine@tournament%
   \@figurine@award%
   \@figurine@dedic%
   \@figurine@pieces%
   \@figurine@stip%
   \@figurine@twins%
   \@figurine@conditions%
   \@figurine@remarks%
   \@figurine@computer%
}
\def\@figurine@number{{\authorfont\thediag)}}
```

```
\def\p@rseauthor@figurine#1,#2; {%
   \if@notfirst, \else\@notfirsttrue\fi#2 #1%
   \l@@klist%
}

\def\@figurine@author{%
   {\ifauth@r%
      \authorfont\@notfirstfalse%
      \let\@action=\p@rseauthor@figurine%
      \@parseTokenlist\aut@tk;%
      \ \ %
   \fi}%
}

\def\@figurine@city{%
   {\if@city%
      \cityfont\@notfirstfalse%
      \let\@action=\p@rsecity\@parseTokenlist\city@tk;%
      \ \ \ %
   \fi}%
}

\def\@figurine@after{\if@after{\dedicfont\ \ \the\after@tk}\fi}

\def\@figurine@correction{%
   \if@correction{\dedicfont\ \ \the\correction@tk}\fi%
}

\def\@figurine@version{%
   \if@version{\dedicfont\ \ \the\version@tk}\fi%
}

\def\@figurine@source{%
   {\if@source%
      \sourcefont%
      \if@sourcenr\the\sourcenr@tk\ \fi%
      \the\source@tk%
      \if@date\ \ \fi\@dia@date%
      \if@issue , \the\issue@tk\fi%
      \if@pages , \the\pages@tk\fi%
   \fi}%
}

\def\@figurine@tournament{%
   \if@tournament{\awardfont\ \ \the\tournament@tk}\fi%
}

\def\@figurine@award{%
   \if@award{\awardfont\ \ \the\award@tk}\fi%
}

\def\@figurine@dedic{%
   \if@dedication{\awardfont\ \ \the\dedic@tk}\fi%
}
```

```
\def\show@squares#1\e@list{\ch@fig{\the\help@a}#1, }

\def\@figurine@pieces{%
   {\if@pieces%
      \let\@action=\p@rsepieces%
      \let\piece@job\show@squares%
      \@parseTokenlist\pieces@tk,%
   \fi}%
}
\def\@figurine@stip{%
   \if@stipulation{\stipfont\ \ \the\stipulation@tk}\fi%
}

\def\@figurine@conditions{%
   \if@condition{\remfont\ \ \the\condition@tk}\fi%
}

\def\@figurine@twins{%
   \if@twins{\remfont\ \ \the\twins@tk}\fi%
}

\def\@figurine@computer{%
   \ifthenelse{\boolean{showcomputer}}{%
      \if@computer\ \computerproofedsymbol\fi%
   }{}%
}

\def\@figurine@remarks{%
   \if@remark{\stipfont\ \ \the\remark@tk}\fi%
}
\def\do@dia@job{\@write@sol\ifvmode\noindent\fi\unhbox\dia@box}
\def\solhead#1{{\split@param{#1}\@dia@solution}}
\def\@write@sol{%
   \ifs@lu%
      \immediate\write\s@lfd{%
         \noexpand\solhead{%
            {\the\label@tk}%
            {\thediag}%
            {\the\aut@tk}%
            {\the\city@tk}%
            {\the\sourcenr@tk}%
            {\the\source@tk}%
            {\the\day@tk}%
            {\the\from@month}%
            {\the\to@month}%
            {\the\year@tk}%
            {\the\issue@tk}%
            {\the\pages@tk}%
            {\the\tournament@tk}%
            {\the\award@tk}%
            {\the\after@tk}%
            {\the\version@tk}%
            {\the\correction@tk}%
            {\the\dedic@tk}%
```

```
            {\the\theme@tk}%
            {\the\twins@tk}%
            {\the\computer@tk}%
            {\the\comment@tk}%
            {\the\judgement@tk}%
            {\the\sol@tk}%
         } %end of \solhead
      }%
   \fi
}
\def\@months#1-#2;{\from@month=#1\to@month=#2\@datetrue}
\def\@dia@writename#1; {\above@newline{\authorfont\@dianame#1; }\l@@klist}
\def\@sol@writename#1; {\sep@names\@solname#1; \l@@klist}
\def\name@sep{,\ }
\def\sep@names{\if@notfirst\name@sep\else\@notfirsttrue\fi}
\def\@checkshort#1/#2#3;{%
   \@shortformtrue%
   \ifx#2\e@list\relax%
      \@shortformfalse%
   \fi%
}
\def\short@christian#1#2-{%
   \if@notfirst -\else\@notfirsttrue\fi%
   #1.%
   \l@@klist%
}

\def\@write@christian#1/#2;{#1}

\def\write@christian#1;{%
   \@checkshort#1/\e@list;%
   \if@shortform\@write@christian#1;\else#1\fi%
}

\def\@write@short#1/#2;{#2}

\def\write@short#1;{%
   \@checkshort#1/\e@list;%
   \if@shortform%
      \@write@short#1;%
   \else%
      {\@notfirstfalse\let\@action\short@christian\l@@klist#1-\e@list}%
   \fi%
}
\def\@fullname#1, #2; {\write@christian#2; #1}
\def\@sirname#1, #2; {#1}
\def\@short#1, #2; {\write@short#2;\ #1}
\def\@noname#1, #2; {}
\def\@normalname#1; {#1}
\def\space@vertical{\space@verticaltrue}
\def\space@horizontal{\space@verticalfalse}
\def\cl@arsol{\immediate\openout\s@lfd=\jobname.sol\relax}
\def\getc@lor#1{%
   \if#1\ds@white%
```

```
      \help@a\z@\global%
      \let\cpd@stepcounterPieces\cpd@stepcounterWhite%
    \else\if#1\ds@neutral%
      \help@a=6\global%
      \let\cpd@stepcounterPieces\cpd@stepcounterNeutral%
    \else\if#1\ds@black%
      \help@a=12\global%
      \let\cpd@stepcounterPieces\cpd@stepcounterBlack%
    \else\errmessage{invalid color!}%
    \fi\fi\fi%
    \getpi@ce%
}

\def\get@text#1{\text@tk={#1}\read@square}

\def\getpi@ce#1{%
    \IfStrEqCase{#1}{%
        {\ds@pawn}{}%
        {\ds@knight}{\advance\help@a\@ne}%
        {\ds@bishop}{\advance\help@a\tw@}%
        {\ds@rook}{\advance\help@a\thr@@}%
        {\ds@queen}{\advance\help@a\f@ur}%
        {\ds@king}{\advance\help@a 5}%
        {C}{%
            \let\cpd@stepcounterPieces\relax
            \advance\help@a 145%
        }%
        {E}{\advance\help@a 216}%
        {X}{\advance\help@a 180}%
    }[\errmessage{invalid piece!}]%
    \futurelet\r@tate\chkr@tate%
}

\def\chkr@tate{%
%     \IfStrEqCase{\r@tate}{%
%         {\ds@rotation@upsidedown}{%
%             \advance\help@a 108%
%             \let\nextpr@c=\skipr@t%
%         }%
%         {\ds@rotation@left}{%
%             \advance\help@a 36%
%             \let\nextpr@c=\skipr@t%
%         }%
%         {\ds@rotation@right}{%
%             \advance\help@a 72%
%             \let\nextpr@c=\skipr@t%
%         }%
%     }[\let\nextpr@c\piece@job]%
    \if\r@tate \ds@rotation@upsidedown\advance\help@a 108\let\nextpr@c=\skipr@t\else%
    \if\r@tate \ds@rotation@left\advance\help@a 36\let\nextpr@c=\skipr@t\else%
    \if\r@tate \ds@rotation@right\advance\help@a 72\let\nextpr@c=\skipr@t\else%
    \let\nextpr@c\piece@job\fi\fi\fi%
    \nextpr@c%
}
```

```
\def\skipr@t#1{\piece@job}
\def\l@@k{\futurelet\whatsnext\parsefi@lds}
\def\parsefi@lds{%
   \if\whatsnext\e@list%
      \let\nextpr@c\relax%
   \else
      \let\nextpr@c\read@square%
   \fi%
   \nextpr@c%
}

\def\set@current@square@index#1#2{%
  \setcounter{cpd@current@square@index}{#1+\value{cpd@lines}*#2}%
}
\def\set@current@square@value#1{%
  \expandafter%
    \xdef\csname cpd@square@\roman{cpd@current@square@index}\endcsname{#1}%
}
\def\get@current@square@value{%
  \setcounter{cpd@current@square@value}%
    {\csname cpd@square@\roman{cpd@current@square@index}\endcsname}%
}
\def\@cpd@setpiece{%
    \ifthenelse{\value{cpd@plane}=\value{cpd@current@plane}}{%
      \cpd@stepcounterPieces%
      \set@current@square@index{\value{cpd@line}}{\value{cpd@row}}%
      \get@current@square@value%
      \ifthenelse{\value{cpd@current@square@value}=\m@ne}%
        {\set@current@square@value{\the\help@a}}%
        {\ifthenelse{\value{cpd@current@square@value}=144}%
          {\set@current@square@value{\the\help@a+18}}%
          {\errmessage{Trying to set a piece to an occupied square.}}}%
    }{}%
}
\def\set@piece{%
    \@cpd@setpiece%
    \l@@k%
}
\def\set@nofield, {%
    \ifthenelse{\value{cpd@plane}=\value{cpd@current@plane}}{%
      \set@current@square@index{\value{cpd@line}}{\value{cpd@row}}%
      \get@current@square@value%
      \ifthenelse{\value{cpd@current@square@value}=\m@ne}%
        {}% This is an empty white square, nothing to do
        {\ifthenelse{\value{cpd@current@square@value}=144}%
          {\set@current@square@value{\m@ne}}%
          {\errmessage{Trying to set a piece to an occupied square}}}%
    }{}%
    \l@@klist%
}
\def\set@frame, {%
    \ifthenelse{\value{cpd@plane}=\value{cpd@current@plane}}{%
      \@vGrid{\value{cpd@line}}{\value{cpd@row}}\@ne%
      \@hGrid{\value{cpd@line}}{\value{cpd@row}}\@ne%
```

```
      \addtocounter{cpd@line}{1}%
      \@vGrid{\value{cpd@line}}{\value{cpd@row}}\@ne%
      \addtocounter{cpd@line}{-1}%
      \addtocounter{cpd@row}{1}%
      \@hGrid{\value{cpd@line}}{\value{cpd@row}}\@ne%
    }{}%
    \l@@klist%
}
\def\e@list{\relax}
\def\l@@klist{\futurelet\nextlist\ch@cklst}
\def\ch@cklst{%
   \ifx\nextlist\e@list%
      \let\nextpr@c=\relax%
   \else%
      \let\nextpr@c=\@action%
   \fi%
   \nextpr@c%
}
\def\@cpd@handle@fen#1{%
   \ifx#1/\relax%
      \ifthenelse{\value{cpd@line}=8}%
      {%
         \setcounter{cpd@line}{0}%
         \addtocounter{cpd@row}{\m@ne}%
      }%
      {%
         \errmessage{FEN: there is now row to end here.}%
      }%
   \fi% FEN space issue
   \IfSubStr{12345678}{#1}{%
      \addtocounter{cpd@line}{#1}%
   }{}%
   \IfSubStr{KQRBNP}{#1}{%
      \let\cpd@stepcounterPieces\cpd@stepcounterWhite%
      \IfStrEqCase{#1}{%
          {K}{\help@a=5}%
          {Q}{\help@a=4}%
          {R}{\help@a=3}%
          {B}{\help@a=2}%
          {N}{\help@a=1}%
          {P}{\help@a=0}%
      }
      \@cpd@setpiece%
      \addtocounter{cpd@line}{\@ne}%
   }{}%
   \IfSubStr{kqrbnp}{#1}{%
      \let\cpd@stepcounterPieces\cpd@stepcounterBlack%
      \IfStrEqCase{#1}{%
          {k}{\help@a=17}%
          {q}{\help@a=16}%
          {r}{\help@a=15}%
          {b}{\help@a=14}%
          {n}{\help@a=13}%
          {p}{\help@a=12}%
```

```
      }
      \@cpd@setpiece%
      \addtocounter{cpd@line}{\@ne}%
   }{}%
}
\def\@cpd@parse@fen#1{\@cpd@handle@fen#1\l@@klist}
\def\p@rsepieces#1, {\getc@lor#1\e@list\l@@klist}
\def\p@rsetext#1, {\get@text#1\e@list\l@@klist}
\def\set@text{%
    \ifthenelse{\value{cpd@plane}=\value{cpd@current@plane}}{%
      \raise\value{cpd@row}\sq@width\hbox to \z@{%
         \hskip\value{cpd@line}\sq@width%
         \vbox to \sq@width{\vss%
         \hbox to \sq@width{%
           \hss%
           {\the\text@tk}%
           \hss%
         }\vss}%
         \hss%
      }%
    }{}%
    \l@@klist%
}
\def\p@rseauthor#1; {\sh@wauthor#1;\l@@klist}
\def\read@square#1#2{%
   \setcounter{cpd@line}{`#1-`a}%
   \setcounter{cpd@row}{#2-1}%
   \read@plane%
}
\def\read@plane@normal{\plane@job}

\def\read@plane@stereo{\futurelet\plane@char\get@plane@stereo}

\def\get@plane@stereo{%
   \if\plane@char A%
      \setcounter{cpd@plane}{1}%
      \addtocounter{cpd@row}{-2}%
      \addtocounter{cpd@line}{-2}%
      \let\@plane@job=\skip@plane%
   \else\if\plane@char B%
      \setcounter{cpd@plane}{2}%
      \addtocounter{cpd@row}{-2}%
      \addtocounter{cpd@line}{-2}%
      \let\@plane@job=\skip@plane%
   \else\if\plane@char C%
      \setcounter{cpd@plane}{3}%
      \addtocounter{cpd@row}{-2}%
      \addtocounter{cpd@line}{-2}%
      \let\@plane@job=\skip@plane%
   \else\if\plane@char D%
      \setcounter{cpd@plane}{4}%
      \addtocounter{cpd@row}{-2}%
      \addtocounter{cpd@line}{-2}%
      \let\@plane@job=\skip@plane%
```

```
    \else%
       \setcounter{cpd@plane}{0}%
       \let\@plane@job=\plane@job%
    \fi\fi\fi\fi%
    \@plane@job%
}

\def\skip@plane#1{\plane@job}

\def\read@plane@space#1{%
     \setcounter{cpd@plane}{`#1-`A}%
     \plane@job%
}
\def\@vGrid#1#2#3{%
    \raise#2\sq@width\hbox to \z@{%
       \hskip#1\sq@width\hskip-.5\grid@width%
       \vrule height#3\sq@width width\grid@width\hss%
    }%
}

\def\@hGrid#1#2#3{%
    \raise#2\sq@width\hbox to \z@{%
       \hskip#1\sq@width%
       \vrule width#3\sq@width height .5\grid@width depth%
       .5\grid@width\hss%
    }%
}
\def\@selGrid#1#2, {%
     \ifthenelse{\value{cpd@plane}=\value{cpd@current@plane}}{%
       \if#1h%
          \@hGrid#2%
       \else\if#1v%
          \@vGrid#2%
       \else%
          \errmessage{Wrong GridSelector #1}%
       \fi\fi%
     }{}%
     \l@@klist%
}
\def\@stdgrid{%
    \setbox\plane@box=\vbox{\hbox{%
       \help@a=\tw@%
       \whiledo{\help@a<\value{cpd@lines}}{%
          \@vGrid{\the\help@a}{0}{\value{cpd@rows}}%
          \advance\help@a\tw@%
       }%
       \help@a=\tw@%
       \whiledo{\help@a<\value{cpd@rows}}{%
             \@hGrid{0}{\the\help@a}{\value{cpd@lines}}%
             \advance\help@a\tw@%
       }%
       \box\plane@box
    }}%
}
```

```
\def\ds@xlabel#1{%
   \label@tk={#1}\@labeltrue%
}

\def\@set@label#1;{\ifds@label\label{#1}\fi}
\def\init@vars{%
   \global\s@lufalse
   \ifthenelse{\boolean{@cpd@numbering@global}}{%
      \setboolean{@cpd@numbering@local}{true}%
   }{%
      \setboolean{@cpd@numbering@local}{false}%
   }%
   \setboolean{cpd@checkPieceCounts}{false}%
   \setcounter{cpd@defWhitePieces}{\z@}%
   \setcounter{cpd@defBlackPieces}{\z@}%
   \setcounter{cpd@defNeutralPieces}{\z@}%
   \setcounter{cpd@whitePieces}{\z@}%
   \setcounter{cpd@blackPieces}{\z@}%
   \setcounter{cpd@neutralPieces}{\z@}%
   \setcounter{cpd@line}{0}%
}

\def\clear@board{%
   \ifthenelse{\boolean{allwhite}\and\boolean{switchcolors}}%
      {\errmessage{'allwhite' and 'switchcolors' do not make sense used together.}}%
      {\@whitefield=\m@ne\@blackfield=144}%
   \ifthenelse{\boolean{allwhite}}{\@blackfield=\m@ne}{}%
   \ifthenelse{\boolean{switchcolors}}{\@whitefield=144\@blackfield=\m@ne}{}%
   \setcounter{cpd@row}{0}%
   \whiledo{\value{cpd@row}<\value{cpd@rows}}{%
      \setcounter{cpd@line}{0}%
      \whiledo{\value{cpd@line}<\value{cpd@lines}}{%
         \set@current@square@index{\value{cpd@line}}{\value{cpd@row}}%
         \setcounter{cpd@helper}{%
                \value{cpd@current@plane}%
                +\value{cpd@line}%
                +\value{cpd@row}}%
         \ifthenelse{\isodd{\value{cpd@helper}}}%
           {\set@current@square@value{\@whitefield}}%
           {\set@current@square@value{\@blackfield}}%
        \addtocounter{cpd@line}{\@ne}%
      }%
      \addtocounter{cpd@row}{\@ne}%
   }%
}


\def\put@row#1{%
   \setcounter{cpd@row}{#1}%
   \setcounter{cpd@row@display}{#1 + 1}%
   \setcounter{cpd@line}{0}%
   \hbox{%
      \ifthenelse{\boolean{legend}}{{%
         \llap{\raise .1\sq@width\hbox{\legendfont\arabic{cpd@row@display}\ \ }}%
```

```
      }}{}%
      \if@stereo%
         \ifthenelse{\value{cpd@current@plane}>0}{%
            \ifthenelse{\value{cpd@row} = 3}{%
               \llap{\cpd@boardfont c6\ }%
            }{}%
            % \fi%
         }{}%
         % \fi%
      \fi%
      % \hbox to \z@{\vbox to \sq@width{}}%
      \set@current@square@index{\value{cpd@line}}{#1}%
      \begin{tikzpicture}[x=\sq@width,y=\sq@width]%
          \useasboundingbox (0, 0) rectangle (\value{cpd@lines}, 1);%
          \whiledo{\value{cpd@line} < \value{cpd@lines}}{%
              \get@current@square@value%
              \ifthenelse{\value{cpd@current@square@value}=\m@ne}%
              {}%
              {%
                  \draw (\value{cpd@line}+.5, .5)%
                  node {{\chessfont\char\value{cpd@current@square@value}}};%
              }%
              \addtocounter{cpd@line}{1}%
              \addtocounter{cpd@current@square@index}{1}%
          }%
      \end{tikzpicture}%
   }%
}
\def\cpd@fen@parseTokenlist#1{\expandafter\l@@klist\the#1\e@list}
\def\@parseTokenlist#1#2{\expandafter\l@@klist\the#1#2 \e@list}
\def\@addToPlane#1{%
   \setbox\plane@box=\vbox{\hbox{%
      \@parseTokenlist#1,%
      \box\plane@box%
   }}%
}
\def\put@plane{%
    % We might want gridchess
    \if@stdgrid%
        \@stdgrid%
    \fi%
    % Let us first set the fieldframes
    \if@fieldframe%
        \let\@action\read@square%
        \let\plane@job\set@frame%
        \@addToPlane\fieldframe@tk%
    \fi%
    % Now we set text to all squares which are given using \fieldtext
    \if@fieldtext%
        \let\@action\p@rsetext%
        \let\plane@job\set@text%
        \@addToPlane\fieldtext@tk%
    \fi%
    % Then we should add the gridlines
```

```
\if@gridlines%
    \let\@action\read@plane%
    \let\plane@job\@selGrid%
    \@addToPlane\gridlines@tk%
\else%
    \if@stereo%
        \stereo@center%
    \fi%
\fi%
% In an 'allwhite' diagram we display dotted lines
\ifthenelse{\boolean{allwhite}}{%
    \setbox\plane@box=\vbox{\hbox{%
        \begin{tikzpicture}[x=\sq@width,y=\sq@width]%
            \useasboundingbox
                (0, 0) rectangle (\value{cpd@lines}, \value{cpd@rows});
        \setcounter{field@border}{1}%
        \whiledo{\value{field@border}<\value{cpd@lines}}{%
            \draw[densely dotted]
                (\value{field@border},0)
                --
                (\value{field@border},\value{cpd@rows});
            \addtocounter{field@border}{\@ne}%
        }%
        \setcounter{field@border}{1}%
        \whiledo{\value{field@border}<\value{cpd@rows}}{%
            \draw[densely dotted]
                (0, \value{field@border})
                --
                (\value{cpd@lines}, \value{field@border});
            \addtocounter{field@border}{\@ne}%
        }%
        \end{tikzpicture}%
        \box\plane@box%
    }}%
}{}%
% Now we should clear the board
\clear@board%
% Let us now parse the list of pieces
\ifthenelse{\boolean{@cpd@fen}}{%
    \ifthenelse{\value{cpd@rows}=8}{}{\errmessage{FEN is only allowed for 8x8 boards.}}% FEN s
    \ifthenelse{\value{cpd@lines}=8}{}{\errmessage{FEN is only allowed for 8x8 boards.}}% FEN
    \setcounter{cpd@row}{7}%
    \setcounter{cpd@line}{0}%
    \let\@action\@cpd@parse@fen%
    \cpd@fen@parseTokenlist\fen@tk\e@list% FEN space issue
}{}%
\if@pieces%
    \let\@action\p@rsepieces%
    \let\piece@job\l@@k\let\plane@job\set@piece%
    \@parseTokenlist\pieces@tk,%
\fi%
% Now we clear all fields, which are given using \nofields
\if@nofields%
    \let\@action\read@square%
```

```
        \let\plane@job\set@nofield%
        \@parseTokenlist\nofields@tk,%
    \fi%
    % Now we can put the pieces to the board
    \global\setbox\plane@box=\hbox{%
        \vbox{\rlap{\box\plane@box}}%
        \vbox{%
           \chessfont%
           \baselineskip=\z@\lineskip=\z@%
           \setcounter{cpd@row}{\value{cpd@rows} - 1}%
           \whiledo{\value{cpd@row} > -1}{%
              \put@row{\value{cpd@row}}%
              \addtocounter{cpd@row}{-1}%
           }%
        }%
        % Put a legend if wanted
        \ifthenelse{\boolean{legend}}{%
            \vbox to \z@{%
                \vbox to \z@{\vss}%
                \llap{\hbox{\hspace*{\inner@frame}%
                    \setcounter{cpd@line}{0}%
                    \setcounter{cpd@line@display}{`a}%
                    \whiledo{\value{cpd@line}<\value{cpd@lines}}{%
                        \hbox to \sq@width{\hfill%
                            {\legendfont\char\value{cpd@line@display}}%
                        \hfill}%
                        \addtocounter{cpd@line}{1}%
                        \addtocounter{cpd@line@display}{1}%
                    }
                }}%
            \vss}%
        }{}%
    }%
}
\def\put@sqs@normal{%
    \put@plane%
    \setbox\sq@box=\hbox{%
       \inner@henbox{\box\plane@box}%
    }%
}
\def\put@sqs@stereo{%
    \setbox\sq@box=\hbox{\hfil\vbox{%
       \vskip\v@space@dist%
       \setcounter{cpd@current@plane}{4}%
       \whiledo{\value{cpd@current@plane}>-1}{%
          \ifthenelse{\value{cpd@current@plane}=0}{%
             \setcounter{cpd@lines}{8}%
             \setcounter{cpd@rows}{8}%
          }{%
             \setcounter{cpd@lines}{4}%
             \setcounter{cpd@rows}{4}%
          }%
          % Now we should clear the board
          \begingroup% We need this for inner loops!
```

```
            \clear@board%
            \put@plane%
         \endgroup%
         \hbox to \bd@width{%
            \hfil%
            \inner@henbox{\box\plane@box}%
            \ifcase\value{cpd@current@plane}\or%
               \rlap{{\cpd@boardfont\ A}}\or%
               \rlap{{\cpd@boardfont\ B}}\or%
               \rlap{{\cpd@boardfont\ C}}\or%
               \rlap{{\cpd@boardfont\ D}}%
            \fi%
            \hfil%
         }%
         \vskip\v@space@dist%
         \addtocounter{cpd@current@plane}{-1}%
      }%
   }\hfil}%
}

\def\stereo@center{%
   \ifthenelse{\value{cpd@current@plane}=0}{%
      \setbox\plane@box=\vbox{\hbox{%
         \@hGrid\tw@\tw@\f@ur\@hGrid\tw@ 6\f@ur%
         \@vGrid\tw@\tw@\f@ur\@vGrid6\tw@\f@ur%
         \box\plane@box%
      }}%
   }{}%
}
\def\put@sqs@space@vertical{%
    \setbox\sq@box=\hbox{\hfil\vbox{%
        \setcounter{cpd@current@plane}{\value{cpd@planes}-1}%
        \vskip\v@space@dist%
        \whiledo{\value{cpd@current@plane}>-1}{%
            \begingroup% We use inner loops!
            \clear@board%
            \put@plane%
            \hbox to \bd@width{%
                \inner@henbox{\box\plane@box}%
                \setcounter{cpd@current@plane@display}{`A+\value{cpd@current@plane}}%
                \rlap{{\cpd@boardfont\ \char\value{cpd@current@plane@display}}}%
            }%
            \endgroup%
            \vskip\v@space@dist%
            \addtocounter{cpd@current@plane}{-1}%
        }%
    }\hfil}%
}

\def\put@sqs@space@horizontal{%
   \setbox\sq@box=\hbox{%
      \setcounter{cpd@current@plane}{0}%
      \hskip\h@space@dist%
      \whiledo{\value{cpd@current@plane}<\value{cpd@planes}}{%
```

```
          % Now we should clear the board
          \begingroup% We use inner loops!
          \clear@board%
          \put@plane%
          \hbox to \bd@width{%
             \inner@henbox{\box\plane@box}%
             \setcounter{cpd@current@plane@display}{`A+\value{cpd@current@plane}}%
             \rlap{{\cpd@boardfont\ \char\value{cpd@current@plane@display}}}%
          }%
          \endgroup%
          \hskip\h@space@dist%
           \addtocounter{cpd@current@plane}{1}%
       }%
    }%
}

\def\put@sqs@space{%
    \ifspace@vertical%
       \put@sqs@space@vertical%
    \else%
       \put@sqs@space@horizontal%
    \fi%
}
\def\@inner@vframe{%
    \if@vframe%
       \vrule width \inner@frame%
    \else%
       \hskip\inner@frame%
    \fi%
}

\def\@inner@hframe{%
    \if@hframe%
       \hrule height \inner@frame%
    \else%
       \vskip\inner@frame%
    \fi%
}
\def\inner@v@frame@rule{%
    \if@stereo%
       \@inner@vframe%
    \else\if@space%
       \@inner@vframe%
    \else\if@leaveOuter%
       \vrule width \inner@frame%
    \else%
       \@inner@vframe%
    \fi\fi\fi%
}

\def\inner@h@frame@rule{%
    \if@stereo%
       \@inner@hframe%
    \else\if@space%
```

```
      \@inner@hframe%
   \else\if@leaveOuter%
      \hrule height \inner@frame%
   \else%
      \@inner@hframe%
   \fi\fi\fi%
}

\def\inner@henbox#1{%
   \hbox{%
      \inner@v@frame@rule%
      \vbox{\inner@h@frame@rule#1\inner@h@frame@rule}%
      \inner@v@frame@rule%
   }%
}
\def\@outer@vrule{\vrule width \outer@frame}

\def\@outer@hrule{\hrule height \outer@frame}
\def\outer@v@frame@rule{%
   \if@stereo%
      \@outer@vrule%
   \else\if@space%
      \@outer@vrule%
   \else\if@leaveOuter%
      \if@vframe\@outer@vrule\else\hskip\outer@frame\fi%
   \else%
      \@outer@vrule%
   \fi\fi\fi%
}

\def\outer@h@frame@rule{%
   \if@stereo%
      \@outer@hrule%
   \else\if@space%
      \@outer@hrule%
   \else\if@leaveOuter%
      \if@hframe\@outer@hrule\else\vskip\outer@frame\fi%
   \else%
      \@outer@hrule%
   \fi\fi\fi%
}

\def\outer@henbox#1{%
   \outer@h@frame@rule%
   \hbox{%
      \outer@v@frame@rule%
      \ifspace@vertical%
         \hskip\h@frame@dist%
      \fi%
      \vbox{%
         \ifspace@vertical%
            \vskip\v@frame@dist%
         \else%
            \vskip\v@space@dist%
```

```
          \fi%
          #1%
          \ifspace@vertical%
             \vskip\v@frame@dist%
          \else%
             \vskip\v@space@dist%
          \fi%
       }%
       \ifspace@vertical%
          \hskip\h@frame@dist%
       \fi%
       \outer@v@frame@rule%
    }%
    \outer@h@frame@rule%
}
\def\ch@fig#1{%
    \ifvmode\noindent\fi%
    \hbox{\chtextfont\lower.1\fontdimen\tw@\chtextfont\hbox{\char#1}}%
}
\def\@dia@index{%
    \@ifundefined{newindex}%
    {\errmessage{You should add documentstyle-option 'index'}}{}%
}

\def\showlabel#1{%
    \if@develop%
       \raise1ex\hbox{\labelfont#1}\penalty\exhyphenpenalty%
    \fi%
}

\def\@aidxitem#1, #2, #3{%
    \par\medskip#1, \write@christian#2; \dotfill #3%
}

\def\dia@index#1\@sep#2[#3]{\index[#3]{#2|showlabel{#1}}}

\def\parse@aindex#1; {%
    \expandafter\dia@index\the\label@tk\@sep#1[author]\l@@klist%
}

\def\@aindex{%
    \if@aindex%
       \ifnormal@names%
          \errmessage{Cannot create index entries with normalnames}%
       \else\ifauth@r%
          \let\@action=\parse@aindex\@parseTokenlist\aut@tk;%
       \fi\fi%
    \fi%
}

\def\x@sindex#1\@sep{\expandafter\dia@index\the\label@tk\@sep#1[source]}

\def\@sindex{%
    \if@sindex\if@source%
```

```
      \expandafter\x@sindex\the\source@tk\@sep%
   \fi\fi%
}

\def\parse@tindex#1, {%
   \expandafter\dia@index\the\label@tk\@sep#1[theme]\l@@klist%
}

\def\@tindex{%
   \if@tindex\if@theme%
      \let\@action=\parse@tindex\@parseTokenlist\theme@tk,%
   \fi\fi%
}
\def\@setPieceColor#1#2#3{%
   \gdef\ds@white{#1}\gdef\ds@black{#2}\gdef\ds@neutral{#3}%
}

\def\@setPieceSpec#1#2#3#4#5#6{%
   \gdef\ds@king{#1}\gdef\ds@queen{#2}\gdef\ds@rook{#3}%
   \gdef\ds@bishop{#4}\gdef\ds@knight{#5}\gdef\ds@pawn{#6}%
}

\def\@setPieceRotation#1#2#3{%
   \gdef\ds@rotation@left{#1}\gdef\ds@rotation@right{#2}\gdef\ds@rotation@upsidedown{#3}%
}
\newcommand{\cpd@piecedef@rotations}{%
    \def\@theRotation{}\cpd@piecedef@colors{0}%
    \def\@theRotation{\ds@rotation@left}\cpd@piecedef@colors{36}%
    \def\@theRotation{\ds@rotation@right}\cpd@piecedef@colors{72}%
    \def\@theRotation{\ds@rotation@upsidedown}\cpd@piecedef@colors{108}%
}
\newcommand{\cpd@piecedef@colors}[1]{%
    \def\@theColor{\ds@white}\cpd@piecedef@types{#1}{0}%
    \def\@theColor{\ds@neutral}\cpd@piecedef@types{#1}{6}%
    \def\@theColor{\ds@black}\cpd@piecedef@types{#1}{12}%
}
\newcommand{\cpd@piecedef@types}[2]{%
    \def\@thePiece{\ds@pawn}\cpd@piecedef{#1}{#2}{0}%
    \def\@thePiece{\ds@knight}\cpd@piecedef{#1}{#2}{1}%
    \def\@thePiece{\ds@bishop}\cpd@piecedef{#1}{#2}{2}%
    \def\@thePiece{\ds@rook}\cpd@piecedef{#1}{#2}{3}%
    \def\@thePiece{\ds@queen}\cpd@piecedef{#1}{#2}{4}%
    \def\@thePiece{\ds@king}\cpd@piecedef{#1}{#2}{5}%
}
\newcommand{\cpd@piecedef}[3]{%
    \setcounter{cpd@piece@index}{#1+#2+#3}%
    \edef\cpd@piecedef@index{\arabic{cpd@piece@index}}%
    \expandafter\xdef%
    \csname\@theColor\@thePiece\@theRotation\endcsname{%
        \ch@fig{\cpd@piecedef@index}%
    }
}

\elchfont\@fselch
```

```
\defaultelchfont%
\diagnum{\@ne}
%% \figcnttrue
\setboolean{piececounter}{true}
\def\@dianame{\@fullname}
\def\@solname{\@fullname}
\space@verticaltrue
\diagnumbering{arabic}
\def\write@month{\@arabic}%
\diagleft
\cl@arsol
\let\orig@author=\author
\let\orig@day=\day
\let\orig@month=\month
\let\orig@year=\year
\let\orig@label=\label
\DefinePieces{wsn}{KDTLSB}{LRU}
\newdimen\normalboardwidth
\def\setboardwidth{%
    \normalboardwidth=\@ight\fontdimen\tw@\chessfont%
    \advance\normalboardwidth\tw@\inner@frame%
    \advance\normalboardwidth\tw@\h@frame@dist%
    \advance\normalboardwidth\tw@\outer@frame%
}

\setboardwidth

⟨/style⟩
```

# 4 The implementation of cpdparse.sty

The following contains the style file *cpdparse.sty*, which implements generic parsing of lists.

```
⟨*cpdparse⟩
\ProvidesPackage{cpdparse}[2020/12/27]

\def\cpd@parse@list{\futurelet\cpd@parse@lookahead\cpd@parse@check}
\def\cpd@parse@check{%
    \ifx\cpd@parse@lookahead\relax\relax%
        \let\cpd@parse@next=\relax%
    \else%
        \let\cpd@parse@next=\cpd@parse@action%
    \fi%
    \cpd@parse@next%
}
```

¡/cpdparse¿

# Index

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

## D



## E

**J**



**L**



**M**



**N**

**T**



**U**



**V**

**W**



**X**



**Y**



# Change History