The RUBIKCUBE package

RWD Nickalls (dick@nickalls.org)
A Syropoulos (asyropoulos@yahoo.com)

This file describes version 5.0 (2018/02/25)
www.ctan.org/pkg/rubik

Abstract

The RUBIKCUBE package provides LaTeX commands and macros for type-setting Rubik cube (3x3x3) notation, configurations, and rotation sequences using the TikZ graphic language. It is part of the Rubik ‘bundle’.

Contents

1 Introduction
1.1 Requirements .................................. 7
1.2 rubikrotation package ................................ 8
1.3 rubikpatterns .................................. 8
1.4 rubiktwocube .................................. 8
1.5 Copyright .................................. 9

2 Installation
2.1 Generating the RUBIKCUBE files .................. 9
2.2 RubikExamples file ................................ 10
2.3 Placing the files .................................. 10
2.4 Usage .................................. 10

3 Command conventions
3.1 Keywords Rubik and Two .......................... 11
3.2 Environments ................................ 12
3.3 Capital letters ................................. 12
3.4 XYZ argument ordering .......................... 12
3.5 Trailing % on the end of commands ................ 13
3.6 Cubies, cubicles, faces and facelets ................. 13

1
4 Rubik cube coordinates 14
  4.1 Size of cube minipage 14
5 TikZ picture environment 15
6 ShowCube command 16
7 Optimum strategy 17
8 Colour commands 18
  8.1 Colour state of the cube 19
  8.2 RubikFace commands 20
  8.3 RubikSolvedConfig command 21
  8.4 RubikCubeSolved commands 21
  8.5 RubikCubeGrey.. commands 23
  8.6 RubikSlice commands 23
9 Rotation commands 24
  9.1 Typesetting 25
  9.2 Face rotations 25
  9.3 Inner-slice rotations 26
  9.4 Outer-slice rotations 27
  9.5 Wide rotations 27
  9.6 Axis rotations 28
  9.7 Examples 29
  9.8 Backwards compatibility 30
  9.9 Listing of all rotation commands 30
    9.9.1 Randelshofer notation 34
  9.10 The rubikfont 36
10 Draw commands 37
  10.1 Error message 38
  10.2 DrawCubie commands 38
  10.3 textCubie commands 39
  10.4 DrawRubikCube commands 39
  10.5 DrawRubikFace.. commands 41
  10.6 Drawing Sidebars (Face) 41
  10.7 Sidebar parameters 42
  10.8 NoSidebar command 42
  10.9 Drawing Sidebars (Cube) 43
  10.10 DrawRubikFlat commands 44
  10.11 DrawNCube 44
11 ShowSequence command 45
  11.1 Trailing digits 46
12 SequenceBrace commands 47
20.21.4 Using \@join ................................................. 97
20.21.5 Rotation B .................................................. 98
20.21.6 Rotation Bp ............................................... 98
20.21.7 Rotation Bw ............................................... 98
20.21.8 Rotation Bwp .............................................. 98
20.21.9 Rotation Bs ............................................... 99
20.21.10 Rotation Bsp ............................................ 99
20.21.11 Rotation Ba .............................................. 99
20.21.12 Rotation Bap ........................................... 99
20.21.13 Rotation D ................................................ 100
20.21.14 Rotation Dp ............................................. 100
20.21.15 Rotation Dw ............................................. 101
20.21.16 Rotation Dwp .......................................... 101
20.21.17 Rotation Ds ............................................. 102
20.21.18 Rotation Dsp ........................................... 102
20.21.19 Rotation Da ............................................. 103
20.21.20 Rotation Dap .......................................... 103
20.21.21 Rotation E ................................................ 104
20.21.22 Rotation Ep ............................................. 104
20.21.23 Rotation F ................................................ 105
20.21.24 Rotation Fp ............................................. 105
20.21.25 Rotation Fs ............................................. 107
20.21.26 Rotation Fsp ........................................... 107
20.21.27 Rotation Fs ............................................. 107
20.21.28 Rotation Fsp ........................................... 107
20.21.29 Rotation Fa ............................................. 107
20.21.30 Rotation Fap .......................................... 107
20.21.31 Rotation L ................................................. 108
20.21.32 Rotation Lp ............................................. 108
20.21.33 Rotation Lw ............................................. 109
20.21.34 Rotation Lwp .......................................... 109
20.21.35 Rotation Ls ............................................. 110
20.21.36 Rotation Lsp ........................................... 110
20.21.37 Rotation La ............................................. 111
20.21.38 Rotation Lap .......................................... 111
20.21.39 Rotation M ................................................. 112
20.21.40 Rotation Mp ........................................... 112
20.21.41 Rotation R ................................................. 113
20.21.42 Rotation Rp ........................................... 113
20.21.43 Rotation Rw ........................................... 114
20.21.44 Rotation Rwp ......................................... 114
20.21.45 Rotation Rs ........................................... 115
20.21.46 Rotation Rsp .......................................... 115
20.21.47 Rotation Ra ........................................... 116
20.21.48 Rotation Rap ......................................... 116
20.21.49 Rotation S .............................................. 117
20.21.50 Rotation Sp .............................................. 117
20.21.51 Rotation Su .............................................. 117
20.21.52 Rotation Sup ............................................. 118
20.21.53 Rotation Sd .............................................. 118
20.21.54 Rotation Sdp ............................................. 118
20.21.55 Rotation Sl .............................................. 119
20.21.56 Rotation Slp ............................................. 119
20.21.57 Rotation Sr .............................................. 119
20.21.58 Rotation Srp ............................................. 119
20.21.59 Rotation SF .............................................. 120
20.21.60 Rotation Sfp ............................................. 120
20.21.61 Rotation Sb .............................................. 120
20.21.62 Rotation Sbp ............................................. 121
20.21.63 Rotation U .............................................. 121
20.21.64 Rotation Uw .............................................. 121
20.21.65 Rotation Up .............................................. 122
20.21.66 Rotation Uwp ............................................ 122
20.21.67 Rotation Us .............................................. 123
20.21.68 Rotation Usp ............................................. 123
20.21.69 Rotation Ua .............................................. 124
20.21.70 Rotation Uap ............................................. 124
20.21.71 Rotations x and xp ...................................... 125
20.21.72 Rotations y and yp ...................................... 125
20.21.73 Rotations z and zp ...................................... 125
20.21.74 Rotations u and d ...................................... 126
20.21.75 Rotations l and r ...................................... 126
20.21.76 Rotations f and b ...................................... 126
20.22 Face(c) and Face(m) rotations .............................. 127
20.22.1 Rotations Rc and Rcp .................................... 127
20.22.2 Rotations Lc and Lcp .................................... 128
20.22.3 Rotations Uc and Ucp .................................... 128
20.22.4 Rotations Dc and Dcp .................................... 128
20.22.5 Rotations Fc and Fcp .................................... 128
20.22.6 Rotations Bc and Bep .................................... 129
20.22.7 Rotations Rm and Rmp .................................. 129
20.22.8 Rotations Lm and Lmp .................................. 130
20.22.9 Rotations Um and Ump .................................. 130
20.22.10 Rotations Dm and Dmp ................................ 131
20.22.11 Rotations Fm and Fmp ................................ 131
20.22.12 Rotations Bm and Bmp ................................ 132
20.23 Randelshofer superset ENG rotations (CMST) .......... 132
20.23.1 Rotations CR and CRp .................................. 133
20.23.2 Rotations CL and CLp .................................. 133
20.23.3 Rotations CU and CUp .................................. 134
20.23.4 Rotations CD and CDp .................................. 134
20.23.5 Rotations CF and CFp .................................. 134
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.23.6</td>
<td>Rotations CB and CBp</td>
<td>135</td>
</tr>
<tr>
<td>20.23.7</td>
<td>Rotations MR and MRp</td>
<td>135</td>
</tr>
<tr>
<td>20.23.8</td>
<td>Rotations ML and MLp</td>
<td>135</td>
</tr>
<tr>
<td>20.23.9</td>
<td>Rotations MU and MUp</td>
<td>136</td>
</tr>
<tr>
<td>20.23.10</td>
<td>Rotations MD and MDp</td>
<td>137</td>
</tr>
<tr>
<td>20.23.11</td>
<td>Rotations MF and MFp</td>
<td>137</td>
</tr>
<tr>
<td>20.23.12</td>
<td>Rotations MB and MBp</td>
<td>138</td>
</tr>
<tr>
<td>20.23.13</td>
<td>Rotations T (wide)</td>
<td>138</td>
</tr>
<tr>
<td>20.23.14</td>
<td>Rotations SR and SRp (opposite slices)</td>
<td>141</td>
</tr>
<tr>
<td>20.23.15</td>
<td>Rotations SL and SLp</td>
<td>141</td>
</tr>
<tr>
<td>20.23.16</td>
<td>Rotations SU and SUp</td>
<td>142</td>
</tr>
<tr>
<td>20.23.17</td>
<td>Rotations SD and SDp</td>
<td>142</td>
</tr>
<tr>
<td>20.23.18</td>
<td>Rotations SF and SFp</td>
<td>143</td>
</tr>
<tr>
<td>20.23.19</td>
<td>Rotations SB and SBp</td>
<td>144</td>
</tr>
<tr>
<td>20.24</td>
<td>Axis rotations (textRubik versions)</td>
<td>144</td>
</tr>
</tbody>
</table>
1 Introduction

The RUBIKCUBE package (part of the RUBIK ‘bundle’) provides a collection of \LaTeX commands and macros for typesetting Rubik cube (3x3x3) configurations using the PGF/TikZ graphic languages. We have extended the rotation hieroglyph notation, originally developed by Garfath-Cox (1981), and improved by Duvoid (2010, 2011).

The RUBIKCUBE package is the ‘base’ package of the bundle, and is required by all of the Rubik packages; it deals primarily with typesetting 3x3x3 cube (Rubik cube) configurations. The RUBIKROTATION package processes rotation sequences and keeps track of the cube’s configuration during rotations. The RUBIKPATTERNS package is a small database of 3x3x3 (Rubik) cube rotation sequences which generate well-known named cube configurations (patterns). The RUBIKTWOCUBE package allows the typesetting of 2x2x2 cube (Two cube) configurations.

Full 3x3x3 functionality requires the following packages to be loaded (TikZ first; RUBIKCUBE second), as follows:

\usepackage{tikz}
\usepackage{rubikcube,rubikrotation,rubikpatterns}

Full 2x2x2 functionality requires the RUBIKTWOCUBE package in addition to the packages listed above. Note that the TikZ package must be loaded before the RUBIKCUBE package.

The RUBIKCUBE package has been road-tested on a Microsoft platform (with MiKTeX), a GNU-Linux platform (Debian 8.2.0 and TeXLive 2017), and on a Solaris platform (OpenIndiana).

For the mathematics and group theory associated with the Rubik cube see Chen (2004), Davis (2006), Fung website, Golomb (1981, 1982), Hofstadter (1981), Hutchings (2011), Heise website, Joyner (2008), Kociemba website, Rokicki et al. (2013), Scherphius website, Tran (2005). Other useful websites are the Speedsolving website, and those maintained by Duvoid, by Fridrich, by Jelinek, by Reid, and by Vandenburgh. A useful online solver utility (based on an algorithm by Kociemba) is available at the RuWix website. Websites with good pages on patterns and symmetries are those by Fridrich, Kociemba, Longridge, Reid, Randelshofer, Scherphius (see References for details).

For historical and technical details regarding Rubik’s cube see Sher (2014), and also the Wikipedia article Rubik’s Cube.

1.1 Requirements

The RUBIKCUBE package requires the TikZ package, since it makes use of the TikZ picture environment and the \pgfmathsetmacro command. Consequently, the TikZ package must be loaded before the RUBIKCUBE package. The RUBIKROTATION package (see below) requires Perl to be installed.
1.2 Supporting tool—the rubikrotation package

The RUBIKROTATION package (also part of the RUBIK ‘bundle’), is a dynamic extension to the RUBIKCUBE package. It consists of the Perl program rubikrotation.pl and the associated style option rubikrotation.sty. The RUBIKROTATION package implements rotation sequences on-the-fly using a \RubikRotation{⟨rotation-sequence⟩} command. It returns the new state in a form which can then be used by the RUBIKCUBE package. It also returns some useful strings associated with the rotation sequence which can be used by the RUBIKCUBE package—see also Section 11.

Since the \RubikRotation command works by calling the rubikrotation.pl program, it follows that the rubikrotation package requires (a) Perl to be installed, and (b) the \LaTeX engine needs to be run using the \texttt{--shell-escape} command-line option. Those wishing to use \LaTeX will also need to have access to the shellesc package (this can always be downloaded from CTAN directly). See the rubikrotation documentation for further details. See also the examples in the file RubikExamples.pdf.

1.3 Supporting database—rubikpatterns package

The RUBIKPATTERNS.sty file (also part of the RUBIK ‘bundle’) is a small database of some well-known 3x3x3 cube (Rubik cube) rotation sequences, stored as named macros. For example, the ‘fourspot’ and ‘sixspot’ sequences are encoded in this package as follows:

\newcommand{\fourspot}{[fourspot],F2,B2,U,Dp,R2,L2,U,Dp,⟨(12q*, 8f*)⟩}
\newcommand{\sixspot}{[sixspot],U,Dp,R,Lp,F,Bp,U,Dp,⟨(8q*, 8f*)⟩}

These sequences can be processed by name (using the \RubikRotation command which also requires Perl to be installed—see Section 1.2), and then displayed (using the \ShowCube command in conjunction with various \DrawRubikCube... commands). So, for example, one could typeset the so-called ‘fourspot’ configuration using the following code:

\usepackage{tikz,rubikcube,rubikrotation,rubikpatterns}
...
\RubikCubeSolved
\RubikRotation{\fourspot} \% this runs the Perl program \texttt{rubikrotation.pl}
\ShowCube{2.4cm}{0.6}{\DrawRubikCubeRU}

The sequence itself can be readily typeset using the \ShowSequence command (see Section 11). See also the RUBIKROTATION documentation—especially Section 5.1.1 Sequences as macros. See also the examples in the file RubikExamples.pdf.

1.4 Supporting 2x2x2 package—rubiktwocube package

The RUBIKTWOCUBE package carries the macros and commands necessary for processing and displaying 2x2x2 cubes (TwoCubes). The 2x2x2 commands are isomorphic with the 3x3x3 commands—i.e., the word ‘Two’ has replaced the word
‘Rubik’ in commands. Consequently, users of this package will need to be familiar with the RUBIKCUBE package. There are lots of 2x2x2 examples in the file RubikExamples.pdf.

1.5 Copyright

Copyright 2014–2018 RWD Nickalls and A Syropoulos.

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3c of this license or any later version. The latest version of this licence is in www.latex-project.org/lppl.txt

2 Installation

The Rubik bundle consists of the four packages RUBIKCUBE, RUBIKROTATION, RUBIKPATTERNS and RUBIKTWOCUBE. Although installing the Rubik bundle will typically install everything automatically (e.g., from the \TeX{}Live DVD), each package can be installed separately if necessary. Here we detail only the RUBIKCUBE package.

2.1 Generating the rubikcube files

Place the file rubikcube.zip into a temporary directory, and unzip it. This will generate the following files:

rubikcube.ins
rubikcube.dtx
rubikcube.pdf --documentation of the rubikcube package
rubik-doc-figA.pdf
rubik-doc-figB.pdf
rubik-doc-figC.pdf
rubik-doc-figD.pdf
rubik-doc-figE.pdf
rubik-doc-figF.pdf
rubikexamples.tex
rubikexamples.pdf
rubikexamples.sh
rubikexamples.bat

The style option rubikcube.sty is generated by running (pdf)\LaTeX{} on the file rubikcube.ins as follows:

pdflatex rubikcube.ins

This documentation file (rubikcube.pdf) can then be generated using the following steps\textsuperscript{1}:

\textsuperscript{1}Several \texttt{pdflatex} runs are required, since the documentation includes an index as well as \texttt{hyperef} links (the package \texttt{hyperdoc} is used). Prior to the first run it is a good idea to delete any relevant \texttt{.toc}, \texttt{.aux}, \texttt{.out} files.
2.2 RubikExamples file

Note that the package includes a ‘rubikexamples’ file (rubikexamples.pdf), as well as the source file (rubikexamples.tex), and associated .sh (Linux) and .bat (Microsoft) batch files, which can be used to facilitate processing the source .tex file. The file rubikexamples.pdf showcases both 3x3x3 (Rubik cube) and 2x2x2 (Two cube) examples.

Note that should you need to generate the file rubikexamples.pdf from the source file (rubikexamples.tex) you will require the rubikrotation, rubikpatterns and rubiktwocube packages to be installed, and will also need to use the --shell-escape command-line option (see Section 1.2 for details).

2.3 Placing the files

Place the files either in the local working directory, or where your system will find them. For a Linux system with a standard TeX Directory Structure (TDS), then:

*.sty → /usr/local/texlive/texmf-local/tex/latex/rubik/
*.pdf → /usr/local/texlive/texmf-local/doc/rubik/

Finally, (depending on your system) update the TeX file database. For example, on a Linux system one uses the texhash command.

2.4 Usage

Load the package by using the command \usepackage{rubikcube}. Note that the RUBIKCUBE package requires the TikZ package, and so always load TikZ before RUBIKCUBE as follows:

\usepackage{tikz}
\usepackage{rubikcube,rubikrotation,rubikpatterns,rubiktwocube}

However, the RUBIKCUBE package does check for the presence of TikZ, and will load it if TikZ is not already loaded.

While RUBIKCUBE is a stand-alone package, for full 3x3x3 functionality it is necessary to load the complementary packages RUBIKROTATION, RUBIKPATTERNS. For full 2x2x2 functionality you need to load all four packages.
3 Command conventions

All RUBIKCUBE package commands assume a 3x3x3 cube by default. Since all cubes are displayed or ‘drawn’ using the TikZ picture environment, it is useful (initially at least) to categorise commands with regard to this environment (and also with regard to the \ShowCube.. command since this is simply a convenient wrapper for the TikZ picture environment). On this basis, we can distinguish three conceptually useful categories, as follows:

1. \Draw.. commands (which must always be used inside a TikZ picture environment),
2. ‘parameter-allocation’ commands (which can be used either inside or outside a TikZ environment); for example, \RubikFace.. (for allocating facelet colours), and
3. commands which can be used in ordinary text; for example, \rr{} (for typesetting certain rotation codes).

From a functional point of view, however, we can view the Rubik bundle commands as splitting into the following groups:

1. those that allocate colour to faces, facelets etc., —these commands all start with \Rubik (for 3x3x3 cubes) or \Two (for 2x2x2 cubes),
2. those that draw —these commands all start with \Draw,
3. those that typeset rotation codes or hieroglyphs; —there are just four of these for 3x3x3 cubes (these commands start with \rr, \rrh, \Rubik, and \textRubik), and an equivalent four commands for 2x2x2 cubes (these start with \tr, \trh, \Two, and \textTwo).

\rubikcube This command generates the logo RUBIKCUBE.

3.1 The keywords Rubik and Two in commands

In order to try and keep commands intuitive\footnote{Requires the RUBIKTWOCUBE package} we adopt the convention that the word ‘Rubik’ in a command reflects the fact that the command relates to a 3x3x3 cube (i.e., a ‘Rubik’ cube). Similarly, commands which relate to a 2x2x2 cube (a ‘Two’ cube) —see the RUBIKTWOCUBE package— use instead the word ‘Two’. For example, the commands for drawing a 3x3x3 cube and a 2x2x2 cube from a RU viewpoint are respectively \DrawRubikCubeRU and \DrawTwoCubeRU.

Having packages now for both 3x3x3 and 2x2x2 cubes (v5) means we need to be more careful regarding command names, and try to make commands (a) as

\footnote{This is a tricky problem given the large number of commands, so any feedback or ideas on how to avoid ambiguity, including pruning or revising ‘bad’ commands, is always welcome.}
intuitive as possible, and (b) use the same command name format for equivalent 3x3x3 and 2x2x2 commands (as shown in the example above).

In keeping with this approach, some commands have had to be renamed. For example, in this new version we have therefore renamed the earlier \DrawFace.. commands → \DrawRubikFace.. (see Section 15).

3.2 Environments

Although the RUBIKCUBE package has been designed with TikZ in mind, it is important to appreciate that of all the various RUBIKCUBE package commands only the Rubik \Draw... commands and TikZ commands actually have to be used inside a TikZ picture environment.

Indeed, using RUBIKCUBE package commands which influence the Rubik colour state (configuration) outside the tikzpictue, minipage or figure environments can make for useful flexibility when a document is generating more than one figure or image. This is because the scope of any colours specified by commands inside these environments is constrained to be ‘local’ to that particular environment, and hence any change in the Rubik colour state brought about by such commands is not accessible globally (i.e., outside the environment) —see also Section 5 in the documentation of the RUBIKROTATION package.

Consequently users need to be mindful of the environments when drawing sequences of rotations across several figures; for example, keeping commands like \RubikRotation, \RubikFace..., \RubikCubeSolved, outside the environments keeps their effects global (an example of this problem is presented in the file RubikExamples.pdf).

3.3 Capital letters

Virtually all Rubik bundle commands start with a capital letter, primarily to avoid any confusion with TikZ commands (these generally start with lower-case letters). However, each ‘word’ in a Rubik bundle command (except the word ‘text’) also starts with a capital letter, primarily to facilitate readability. For example, \DrawRubikCubeRU, \DrawCubieRU. However, as with LATEX, ‘text,’ commands start with a lower-case ‘t’; for example \textCubieRU. Letter arguments for colours (R, O, Y, G, B, W, X) are always written in upper-case letters.

3.4 XYZ argument ordering

Many commands have an appended two, three, or even six ordered arguments or letters which form some feature of the structure of a command; perhaps either face or colour code or a viewpoint direction.

We adopt the convention that where ordering of arguments is critical, then the arguments are ordered in the XYZ, +, −, order. An XYZ code implies that the first letter in the code relates to an X-related parameter, for example, L (Left) or R (Right); the second letter relates to a Y-related parameter, for example, U (Up)
or D (Down); the third (if required) relates to a Z-related parameter, for example, F (Front) or B (Back) —see Figure 2.

Some commands have six arguments which adopt an (XYZ;+−) format. In this case, for example, the \RubikSolvedConfig command, for which the six colour arguments are ordered as X+, X−, Y+, Y−, Z+, Z−. Here the colour argument associated with a face positioned on the +ve axis is ordered before its −ve complement on the same axis.

Another example is the \DrawCubieRU{G}{Y}{O} command, which draws a cubie. Here the RU letters are XY ordered; i.e., RightUp viewpoint. The sequence of colour codes for the three visible faces are XYZ ordered, and hence result in the cube having a Green Right face, Yellow Up face and Orange Front face.

3.5 Trailing % on the end of commands

Since the all the output of this package is drawn using graphic elements using TikZ, it is important to include a trailing % on the end of RUBIKCUBE package commands when used outside a TikZ picture environment, and also on the end of the \end{tikzpicture} environment command itself. In particular it is important to use a trailing % on the end of lines which break before the terminal curly bracket of a \newcommand.

This is to prevent accumulating spurious spaces which may otherwise appear in figures and diagrams as a strange or unexpected horizontal shift or white-space. That this can occur is because in T\TEX every newline character is automatically converted to a white space—unless you have an empty line (Feuersänger 2016).

The L\TEX fbox is a useful aid for visualising unwanted white space which may have accumulated, and for identifying the cause. See Section 6 on the \ShowCubeF command for more details regarding this approach.

Although this effect is mostly small, and is generally only observed in situations when centering a graphic is critical, it is, however, cumulative and can be surprisingly large. In these situations, the cure is the addition of terminal % characters to preceding code guided by careful detective use of the fbox technique mentioned above.

3.6 Cubies, cubicles, faces and facelets

The sub-cubes which make up the Rubik cube are known as ‘cubies’; the small coloured face of a cubie is known as a ‘facelet’. The cubies are named either according to the colours of their two or three facelets, or according to their physical position.

We distinguish three types of cubie: centre-cubies (single colour), edge-cubies (two colours) and corner-cubies (three colours). For example, the red/white edge-cubie is called the RW cubie, and the red/white/green corner-cubie is called the RWG cubie etc. Note that the colour of a particular face of a 3x3x3 Rubik cube is determined by the colour of its centre-cubie.

Similarly, the positions (known as ‘cubicles’) occupied by cubies are defined using either a two or three letter face code. For example, the right edge position
in the Up-layer is termed the Up/Right position, or just the UR position, and the corner joining the DOWN FRONT and RIGHT faces is the DFR position.

4 Rubik cube coordinates

The coordinate origin of all 2D cube images is located at the bottom-left corner of the FRONT face, as shown in Figure 1. Note also that the bottom left extent of this particular 2D rendering of the 3x3x3 cube is actually at (−1, −1), and hence the default height and width of all oblique-view cubes is 4 units (i.e., equivalent to 4cm if the TikZ scale-factor = 1).

Arranging for $P$ to be at (−1, −1), as well as using the bottom-left corner of the FRONT face as the origin, is a useful design feature which make it easy to figure-out the coordinates of any point on the image plane (either on the cube or outside the cube), and hence facilitates the use of TikZ commands (e.g., \draw and \node commands) to superimpose lines, arrows and text etc., onto the Rubik cube (see Section 13).

4.1 Size of cube \minipage

Since the the default height and width of the oblique 2D cube image is 4 units (see Section 4 above), it follows that the width of the \minipage required for a cube in a tikzpicture environment can be easily calculated. For example, if the tikzpicture scale factor used is 0.5, then the minimum width of the required minipage for the \DrawRubikCubeLD view (shown above) is therefore $0.5 \times 4\text{cm} = 2\text{cm}$.

Note that the width of the semi-flat (SF) cube representation is therefore 10 units ($= 3+3+1+3$), and that of the flat (F) cube is 12 units ($= 3+3+3+3$) — see Section ?? for images of these forms. If in doubt check the horizontal extent of an image using the \ShowCubeF command, which places an fbox around the image.
5 TikZ picture environment

All the Rubik bundle \Draw commands are designed to be used with the TikZ picture environment, and are compatible with standard TikZ. For a basic introduction to the use of TikZ see the following manuals (from CTAN or from http://altermundus.com/).

- pgfmanual.pdf, version 3.0.1a (2015) (1161 pages)
- tkz-base-screen.pdf, version 1.16c (2011) (91 pages)

An example of the TikZ picture environment for use with the Rubik bundle is as follows:

\begin{tikzpicture}[scale=0.5]
...
\end{tikzpicture}

If no scale-factor is used (default scale-factor = 1), then each of the small cubie sides will have a length of 1 cm.

USEFUL COMMANDS: Probably the most useful TikZ commands for use with regard to the Rubik bundle are the \draw command (for drawing lines, arrows, circles), and the \node command (for writing text at specific coordinate locations). The basic structure of these commands is as follows, where (x,y) represent grid coordinates of start or end points of lines or arrows, or of a circle centre, or of text position (see Sections 10.4 and 13 for examples).

\draw[->,thick,color=blue] (4.5, 2.5) -- (3.5,2.5);
\draw[->,ultra thick,color=red] (4.5, 2.5) -- (3.5,2.5);
\draw [color=blue, thick] (0.3, 0.3) circle (1.3);
\node (B) at (7.5, 1.5) [black]{B};

Remember that all TikZ commands which are valid inside a tikzpicture environment require a terminal semicolon (see Section 13 for examples).

COLOURS: The following colors are predefined by TikZ: red, green, blue, cyan, magenta, yellow, black, gray, darkgray, lightgray, brown, lime, olive, orange, pink, purple, teal, violet and white (see https://en.wikipedia.org/wiki/PGF/TikZ).

LINE WIDTH: TikZ allows line width to be specified directly (e.g., [line width=<dimension>]), or by using the following abbreviations: ‘ultra thin’ for 0.1pt, ‘very thin’ for 0.2pt, ‘thin’ for 0.4pt (the default width), ‘semi thick’ for 0.6pt, ‘thick’ for 0.8pt, ‘very thick’ for 1.2pt, ‘ultra thick’ for 1.6pt (see https://en.wikipedia.org/wiki/PGF/TikZ).

WHITE SPACE: A particularly useful feature of TikZ is that it automatically minimises any horizontal white-space. However, it is good practice to place a % symbol after the \end{tikzpicture} command to avoid additional white space inadvertently being added by \LaTeX (see Section 3.5).
When making images it can be helpful to place them inside a minipage (e.g., using the `\ShowCube` command / environment below). A convenient approach is to first adjust the value of the tikzpicture scale-factor (to obtain the appropriate size), and then adjust the minipage-width as necessary, using the fbox associated with the `\ShowCubeF` command (see Section 4.1 for a useful guide on this).

The main ‘display’ tool for drawing cubes is the `\ShowCube` command (see below), and this incorporates a TikZ picture environment inside a minipage. The equivalent tool for displaying rotation sequences is the `\ShowSequence` command.

### 6 \ShowCube command

This command `\ShowCube{⟨width⟩}{⟨scale-factor⟩}{⟨commands⟩}` is a convenient tool for placing one or more commands inside a tikzpicture environment which is also inside a minipage (see Section 20.4 for the code). This command takes three arguments: the first (#1) is the minipage width, the second (#2) is the tikzpicture scale factor, and the third (#3) is a series of any \rubikcube{} package \Draw.. and other commands, as well as any TikZ commands which are valid in a tikzpicture environment (e.g., \draw or \node etc.).

**Usage:** The following `\ShowCube` command displays a Rubik cube (the ‘SixT’s configuration) and a blue arrow in a minipage of width 3cm, using a tikzpicture scale factor of 0.5. Note that the TikZ \draw command requires a terminal semicolon (see Section 5).

```latex
\RubikCubeSolved
\RubikRotation{\sixts}
\ShowCube{3cm}{0.5}{\%
  \DrawRubikCubeLU
  \draw[\rightarrow, thick, color=blue] (4.5, 2.5) -- (3.5, 2.5);
}\%
```

The action of the `\ShowCube` command is illustrated below; the `\ShowCube` command on the left is equivalent to the bunch of commands on the right (see Section 20.4 for the complete code).

```latex
\begin{minipage}{3cm}\%
  \centering
  \begin{tikzpicture}[scale=0.5]
  ...
  \end{tikzpicture}\%
  \end{minipage}\%
```

The `\ShowCubeF` command is similar in all respects except that it places an fbox around the minipage in order to enable users to see the extent of any associated white space. For example, unexpected spacing between two adjacent images, or between an image and adjacent text, is usually related to ‘hidden’ white-space associated with the image itself or excessive width of the associated `\minipage`.

---

^4The \sixts macro is from the \rubikpatterns{} package.
(see also Section 3.5). Consequently, a temporary \texttt{fbox} around the minipage can be a useful aid when trying to visualise the full extent of the minipage (and its associated white-space). Use the \texttt{\ShowCubeF} command for this.

For example, the following use of the \texttt{\ShowCubeF} command reveals a significant white-space problem:

\begin{verbatim}
\ShowCubeF{4cm}{0.3}{\DrawRubikCubeRU}
\end{verbatim}

In this example, clearly either the minipage is too wide (4cm) or the \texttt{tikzpicture} scale factor is too small (0.3). Once the figure/code has been corrected, then the \texttt{F} in the \texttt{\ShowCubeF} command can be removed.

Note that while the \texttt{\ShowCube} command centres the image inside the minipage, \LaTeX{} positions the minipage in the \texttt{textwidth}, and hence it is generally best to minimise the horizontal white-space as revealed by the \texttt{\ShowCubeF} command. The relationship between the required width of the minipage and the TikZ scale factor for the various Rubik cube images is detailed in Section 4.1.

\section{Optimum strategy}

We suggest that the most convenient (and intuitive) approach for drawing cubes or particular faces is to do it in stages, as follows (all these steps are well illustrated in the examples file \texttt{RubikExamples.pdf}):

\begin{itemize}
  \item first, start by setting the colour state of the cube. This can be done using either (a) a \texttt{\RubikCubeSolved}.. or \texttt{\RubikCubeGrey}.. command (for defining the whole cube), or (b) using one or more \texttt{\RubikFace}.. commands (for defining parts of faces), or (c) by imputting a file containing a previously saved colour state\textsuperscript{5}.
  \item second, use the \texttt{\RubikRotation} command to process a sequence of rotations (remembering that this requires use of the \texttt{--shell-escape} command-line option). The \texttt{RUBIKPATTERNS} package is a small library of named rotation sequences.
  \item third, draw the image(s) using \texttt{\DrawRubikCube}.. or \texttt{\DrawRubikFace}.. commands, plus any TikZ commands (e.g., \texttt{\draw} and/or \texttt{\node}) in conjunction with the \texttt{\ShowCube} command. Use the \texttt{\ShowCube} scale factor to adjust the size, and use the \texttt{\ShowCubeF} command to reveal the extent of any \texttt{minipage} whitespace.
  \item fourth, spacing between graphic elements can be influenced by adjusting either (a) horizontal white-space as set by the \texttt{\ShowCube} command, or (b) using standard \LaTeX{} spacing commands, e.g., \texttt{\quad}, \texttt{\qquad}, \texttt{\hspace}.. etc.
\end{itemize}

\textsuperscript{5}See the \texttt{RUBIKROTATION} package documentation for details of the \texttt{\SaveRubikState} command; see also the ‘SaveRubikState’ example in the file \texttt{RubikExamples.pdf}.
• finally, give some thought to using a trailing % in commands which are broken across multiple lines (see Section 3.5).

With this approach the internal colour state will be updated and processed correctly by all subsequent Draw... or RubikRotation commands. Note that exchanging the word ‘Rubik’ for the word ‘Two’ in a command will generate the equivalent TwoCube version of the command (see Section 3.1).

8 Colour commands

The Rubik bundle of packages uses seven colours which are defined as follows: R (red), O (orange), Y (yellow), G (green), B (blue), W (white), and X (grey). Now according to the following webpage


the official Rubik cube colours are defined as

... colours which are red (PMS 200C*), green (PMS 347C*), blue (PMS 293C*), orange (PMS 021C*), yellow (PMS 012C*) and white.

... Pantone colors can not be accurately converted to RGB colors, the colors the web runs on. But they can be approximated. Through some research, I have found some estimations which may help you which I have listed below. Remember, these are just approximate RGB equivalents to the official Rubik’s Cube colors.

Red: 200C #C41E3A (www.perbang.dk/rgb/c41e3a/)
Green: 347C #009E60 (www.perbang.dk/rgb/009e60/)
Blue: 293C #0051BA (www.perbang.dk/rgb/0051ba/)
Orange: 021C "Pantone Orange" #FF5800 (www.perbang.dk/rgb/ff5800/)
Yellow: 012C "Pantone Yellow" #FFD500 (www.perbang.dk/rgb/ffd500/)
White: N/A #FFFFFF

Red  {HTML}{C41E3A}
green {HTML}{009E60}
Blue  {HTML}{0051BA}
Yellow {HTML}{FFD500}
Orange {HTML}{FF5800}
White  {HTML}{FFFFFF}

The following RGB specifications are given by Sher (2014):

White  {RGB}{255,255,255}
Red    {RGB}{137,18,20}

---

6We thank Peter Bartal for bringing this webpage to our attention.
Blue {RGB}{13,72,172}
Orange {RGB}{255,85,37}
Green {RGB}{25,155,76}
Yellow {RGB}{254,213,47}

However, we have tried to optimise these prescribed colours very slightly for screen 
& print use (for example, the yellow was made very slightly brighter), and so 
the actual colours implemented by the RUBIKCUBE package are as follows\footnote{Although the Pantone colours cannot be converted to RGB, there is a subset of of Pantone 
colours which can be converted using CMYK (see https://en.wikipedia.org/wiki/Pantone).} (see 
Section 20.2):

\definecolor{R}{HTML}{C41E33}
\definecolor{G}{HTML}{00BE38}
\definecolor{B}{HTML}{0051BA}
\definecolor{Y}{HTML}{FFFF00}
\colorlet{O}{orange}
\colorlet{W}{white}
\colorlet{X}{black!30}%

Different colours can be allocated to the ROYGBWX letters (using the standard \LaTeX \colorlet command) as required. For example, the standard ‘red’ colour 
could be allocated to the letter R using the command

\colorlet{R}{red}

However, it is important to appreciate that the letter codes ROYGBWX are ‘hard-
wired’ into many of the macros in the RUBIKCUBE package, so don’t change these.

\section{Colour state of the cube}

A given cubic facelet on a given face is denoted using an ordered sequence of three 
letters, as follows: first the face code (U,D,L,R,F,B), second the X-position of the 
column (l,m,r), and third the Y-position of the row (t,m,b). For example, the 
‘right-bottom’ facelet of the FRONT face is denoted as Frb, and consequently the 
current colour-code (R,O,Y,G,B,W,X) of this facelet is held as the variable \Fr{b} etc. (see Section 20.7 for details and code).

Initially, when \LaTeX reads the file rubikcube.sty all facelets are allocated 
the colour-code X, which can be regarded as a zero-colour state. Until a facelet is 
allocated one of the six cube colours (using a suitable command) it will be rendered 
as grey by a \Draw... command, since these commands simply implement the 
current colour state of the cube (e.g., \DrawRubikCubeRU). Facelets retain their 
colour allocation even if they are moved using the \RubikRotation command (see 
RUBIKROTATION package), unless they are overwritten by a subsequent colour 
allocation command.

Colours are allocated to facelets using using \Rubik.. commands. For ex-
ample, the commands \RubikCubeSolvedWY and \RubikCubeSolvedWB allocate 
prescribed colour states for the whole ‘solved’ cube, and are a very useful starting
point (configuration) for subsequent rotations. The commands \RubikCubeGreyWY and \RubikCubeGreyAll allocate different colour states for the whole cube, and are designed to be useful starting points when illustrating aspects of how to solve the cube. These two commands accept both ‘grey’ and ‘gray’ (to be consistent with TikZ).

Colours can also be allocated to subsets of facelets (e.g., faces, slices etc); for example, using the commands \RubikFace... and \RubikSlice... commands (see Sections 8.2 and 8.6).

To visualise the current state of the cube one has to use a \Draw... command. \Draw... commands never influence the internal colour state of the cube.\footnote{That said, the now deprecated \DrawRubikLayerFace... and \DrawRubikLayerSlice... commands (see Section 15) did, confusingly, allow you to specify colours as arguments, but they only ‘painted’ colours onto facet positions (on the page, so to speak), and for this reason they are now deprecated, and will be phased out in due course.}

The current colour state / configuration of a cube can also be saved and written to a named file, which can then be \input and processed later when required, using the \SaveRubikState command (3x3x3 cube) or \SaveTwoState command (2x2x2 cube).

### 8.2 RubikFace commands

These commands allocate colours to the individual cubies of a 3x3x3 cube face; they take nine colour arguments (see Section 20.7 for the code). The ordering is isomorphic to the sequence 1–9, i.e., numbering the small squares 1-3 (top row, left to right), 4-6 (middle row, left to right), 7-9 (bottom row, left to right), as follows:

```
#1 #2 #3
#4 #5 #6
#7 #8 #9
```

Conveniently, \LaTeX allows the colour arguments to be separated by spaces (e.g., in groups of three), or even spread across several lines (e.g., in a square block to resemble a 9-face). This is fortunate, as it allows the command to be written in several visually intuitive ways, as follows:

\begin{verbatim}
RubikFaceUp{G}{B}{G} {G}{W}{O} {G}{O}{G}
RubikFaceFront{O}{W}{R}
{W}{W}{W}
{G}{W}{G}
\end{verbatim}

Failure to include a valid colour argument will generate a ‘missing parameter’ error, and no colour will be allocated (i.e., you will see a black-hole) when it is rendered.

Each of the above commands has an associated ‘All’ version which allocates the same colour to all the cubies on a 9-face (i.e., only a single colour argument is required). For example, if you want the \textit{right} face to be all orange, then use...
the command \RubikFaceRightAll\{O\}. Use of these commands is shown in the following example.

\RubikCubeGreyAll
\RubikFaceRightAll\{O\}
\RubikFaceFront\{W\}{Y}\{G\}
  \{W\}\{Y\}\{G\}
  \{W\}\{Y\}\{G\}
\ShowCube\{3cm\}\{0.7\}\{\DrawRubikCubeRU\}

Note that instead of using \RubikCubeGreyAll we could have used the command \RubikFaceUpAll\{X\} to allocate grey to the whole of the UP face. However, the \RubikCubeGreyAll command can be a useful starting point when dealing with a new cube, since it resets all the faces to their initial default colour.

Finally, it is important to bear in mind that when allocating colours using the \RubikFace\ commands it is very easy to inadvertently create a non-valid cube (ie a cube with either the wrong number of facelets with particular colours, or one which has a non-sovable configuration). However, some basic error checking of this sort is done whenever the \RubikRotation command is used (see the RUBIKROTATION package documentation).

8.3 RubikSolvedConfig command

\RubikSolvedConfig

This command allocates the six face colours according to the following ordered XYZ+- argument rule, namely X+, X-, Y+, Y-, Z+, Z-; i.e., the order of the six colour arguments follows the face order RIGHT, LEFT, UP, DOWN, FRONT, BACK (for notation see Section 3.4 and Figure 2).

Usage: \RubikSolvedConfig\{G\}\{B\}\{W\}\{Y\}\{O\}\{R\}

Examples of its use are shown in the next section.

8.4 RubikCubeSolved commands

The action of both of these commands is identical: they both set all the face colours to the following standard ‘solved’ cube configuration, namely Up=white, Down=yellow, Right=green, Left=blue, Front=orange, Back=red, by invoking the above \RubikSolvedConfig command, as follows:

\newcommand\{\RubikCubeSolved\}{\{\RubikSolvedConfig\{G\}\{B\}\{W\}\{Y\}\{O\}\{R\}\}}

Note that this is in fact just a convenient short-hand for the following:

\newcommand\{\RubikCubeSolved\}{\%
  \RubikFaceRightAll\{G\}\%
  \RubikFaceLeftAll\{B\}\%
  \RubikFaceUpAll\{W\}\%
  \RubikFaceDownAll\{Y\}\%
  \RubikFaceFrontAll\{O\}\%
}
Note that for convenience, this configuration is also available using the command \RubikCubeSolvedWY (WY denoting White opposite Yellow). This solved configuration is shown in the following semi-flat (SF) image.

\begin{figure}[h]
\centering
\RubikCubeSolvedWY
ShowCube{5cm}{0.5}{\DrawRubikCubeSF}
\end{figure}

Note that the width of the minipage used in the \ShowCube command above is set to 5cm. This value is derived from the fact that the unscaled width of the semi-flat image is 10cm (9 + 1 squares), and hence if the TikZ scale factor is set to 0.5 (as in the above example) then the minimum minipage width = 10 \times 0.5 = 5cm (see Section 4 for details).

Other orientations: If other orientations of the solved cube are required, this can be easily achieved using the \RubikRotation command (from the RubikRotation package) to rotate the cube as required. For example, we could make a command to show the above solved cube upside-down and rotated slightly, as follows:

\begin{verbatim}
\newcommand{\CubeUpSideDown}{\RubikCubeSolved\RubikRotation{x2,y}}
\end{verbatim}

—this uses the rotations $x$, $x$, to invert, and then $y$ to turn the cube 90 deg:

\begin{figure}[h]
\centering
\CubeUpSideDown
\ShowCube{2cm}{0.5}{\DrawRubikCubeRU}
\end{figure}

Other configurations: While the ‘solved’ WY colour configuration described above (White face opposite Yellow) is that of the most commonly occurring Rubik cube, another ‘solved’ colour configuration which is also commercially available has the White opposite Blue configuration, which is available using the command \RubikCubeSolvedWB. Its colour configuration is as follows:

\begin{figure}[h]
\centering
\RubikCubeSolvedWB
\end{figure}
Note that users can easily create their own alternative ‘solved’ face/colour versions. For example, the above mentioned white opposite blue (WB) solved configuration command \RubikCubeSolvedWB (white opposite blue, red opposite orange, and green opposite yellow), was created using \RubikSolvedConfig{R}{O}{W}{B}{G}{Y} (for the code see 20.11).

### 8.5 RubikCubeGrey.. commands

The command \RubikCubeGreyWY generates a 3x3x3 cube with no colours allocated except for the central cubie of each face, which takes the same colour configuration as defined for the \RubikCubeSolvedWY command. The command \RubikCubeGreyAll generates a cube with *all* the faces completely grey; this is useful as it can be used to reset all the facelets to the initial default state. These commands will accept either ‘grey’ or ‘gray’ (to be consistent with TikZ).

These commands are designed to be useful starting points when wanting to describe the movement of particular cubies. We can see the effect of the \RubikCubeGreyWY command by viewing the cube in a semi-flat (SF) format, as follows:

```
\RubikCubeGreyWY \ShowCube{4.5cm}{0.45}\DrawRubikCubeSF
```

Users can of course set up their own alternative face/colour configuration by creating a new ‘variant’ command altogether.

### 8.6 RubikSlice commands

These three commands allocate the six visible cubie colours associated with a *horizontal* slice of a Rubik cube. There are three pairs of Slice commands; one pair (Left view & Right view) for each of the horizontal slices Top, Middle, Bottom. The six colour arguments associated with a given slice run in sequence from left to right irrespective of the viewpoint, e.g., #1 #2 #3 #4 #5 #6.

Since the viewpoint of the Rubik cube (from the Right or from the Left) influences which face the colours are associated with, it is necessary to have the view (R or L) specified in the command name.

The format of the ‘slice’ command is shown in the following example. The Rubik cube is shown from the LeftDown (LD) view and consequently each of the ‘slice’ commands in this particular example ends in L, consistent with the final \DrawRubikCubeLD command.

Note that the two legacy ‘Equator’ versions (now replaced by ‘Middle’) are retained to allow backward compatibility.
9 Rotation commands

The Rubik bundle implements not only the standard Rubik cube notation of the World Cube Association (see WCA website), but also the main variant notations used by the Rubik interest groups and websites.

To avoid confusion the Rubik bundle uses a trailing ‘p’ (lower case) in rotation-codes to denote a ‘prime’ (reversed direction); we also recommend that commas are used to separate sequential Rubik rotations (moves). While these are mainly to avoid ambiguity, they also greatly facilitate computer searching and copy-and-pasting of rotation sequences.

Unfortunately, obtaining a good balance between an intuitive notation for defining rotations and the need for flexibility is difficult, and consequently some notation is more intuitive than others. A good compromise seems to be the World Cube Association’s FADN structure; i.e., Face (L,R,U,D,F,B), Action (m,w,s,a,c), Direction (p), N (n); for example, codes like R, R2, Rc, Rm, Rwp, Rwp2 etc.

The RUBIKCUBE package includes commands for typesetting a wide range of rotation-codes (e.g., \( R \), \( y \), \( Bw \)) and equivalent hieroglyphs (e.g., \[ H \], \[ y \], \[ Bw \]), as well as commands for typesetting 3x3x3 cubes\(^9\) and single cubies. All the rotation-codes and hieroglyphs are typeset using one particular font & size which we call the ‘rubikfont’ for convenience (see Section 9.10 for details). All of the

---

\(^9\)See the RUBIKTWOUCUBE package documentation for 2x2x2 cube commands.
rotation-codes described here are recognised by the RUBIKROTATION package (see Section 1.2).

Note that there are some rotation codes which are not represented by arrow hieroglyphs, since their rotation is not visible from the front face, and hence cannot easily be rendered as an arrow hieroglyph. Consequently these rotations have a simple ‘letter’ hieroglyph in the form of the rotation-code in a square; for example, \texttt{Bw}, \texttt{Fm}.

9.1 Typesetting

We now describe the four commands used for typesetting the various rotation-codes.

\texttt{\rr} The text version of a rotation-code is typeset using the rubik-rotation command \texttt{\rr\{rotation-code\}}, i.e., \texttt{R} is typeset using the command \texttt{\rr{R}}. The hieroglyph of a rotation is generated (in text) by using instead the command \texttt{\rrh\{rotation-code\}}. For example, the command \texttt{\rrh{R}} generates \texttt{\textbf{R}}, which is the hieroglyph associated with the rotation \texttt{R}.

\texttt{\Rubik} A vertically combined rotation-code and its hieroglyph is generated using the command \texttt{\Rubik\{rotation-code\}}. For example, \texttt{R} is generated by the command \texttt{\Rubik{R}}, with the square hieroglyph sitting on the baseline. For some hieroglyphs (e.g., \texttt{x}, \texttt{y}, \texttt{z} denoting 90 degree cube-axis rotations) the only difference between the \texttt{\rrh{}{}} and \texttt{\Rubik{}{}} form is that the \texttt{\Rubik{}{}} form is elevated to sit on the baseline just like the other \texttt{\Rubik{}{}} hieroglyphs. For example, \texttt{\rrh{yp}} generates \texttt{\textbf{y}'}, while \texttt{\Rubik{yp}} generates \texttt{\textbf{y}'.}

\texttt{\textRubik} A horizontally combined rotation-code and its hieroglyph (in sequence as in text) is generated using the command \texttt{\textRubik\{\text{rotation-code}\}}. For example, \texttt{R} is typeset using the command \texttt{\textRubik{R}}. A list of all rotation-code commands and their associated hieroglyphs is given in Section 9.9.

9.2 Face rotations

\texttt{U} The six main faces of the cube are denoted as front (towards the observer), back, left, right, up, down. The upper-case initial letter of each face-name \texttt{F}, \texttt{B}, \texttt{L}, \texttt{R}, \texttt{U}, \texttt{D} denotes a clockwise 90-degree rotation of the face as shown in Figure 2. For example, \texttt{D} is generated by the ‘rubik rotation’ command \texttt{\rr\{D\}}.

\texttt{F} An appended prime ‘ indicates an anticlockwise rotation; e.g., \texttt{F'}. This is sometimes written as \texttt{F^{-1}}. The ‘prime’ notation is achieved by appending a lower-case \texttt{p} to the face rotation command. For example, \texttt{R'} is generated by \texttt{\rr\{Rp\}}. More formally, \texttt{R'} is the ‘inverse’ of \texttt{R}.

\texttt{Lp} The superscript \texttt{2}, or sometimes just an ordinary \texttt{2}, indicates that the rotation is applied twice. For example, \texttt{R^2} or \texttt{R2} denote two successive 90 degree clockwise rotations of the right face; \texttt{R^3} is equivalent to \texttt{R'} etc.
9.3 Inner-slice rotations

The Rubik cube (3x3x3) has three orthogonal so-called ‘inner’ slices (middle layers, middle slices), whose +ve rotation direction follows that of a named face. For example, the inner-slice rotation between the RIGHT and LEFT faces whose rotation direction follows the rotation $R$ (i.e., its rotation is isomorphic to $R$). The inner-slice rotations form a group (the Slice group), originally described by John Conway (Frey and Singmaster, 1982, p 105).

The ‘m’ notation

Here ‘m’ stands for the ‘middle’ slice, namely that parallel to the designated face; its rotation mirrors that of the face. The $m$ must be in lower case. Each of these rotation-codes has a complementary ‘prime’ version, formed by appending a ‘p’; for example, $Rm$ (\texttt{Rm}) is a middle layer rotation \texttt{[M]} between the RIGHT and LEFT faces, and is in the same direction as $R$. The code $Rm'$ (\texttt{Rmp}) refers to the same middle slice, but rotated in the opposite direction \texttt{[M]}.

This notation, which was probably invented by Singmaster, was originally used on the Cube Lovers usenet group (1981–1997). It is now much used on the Jaap Puzzles website (see Scherphius J) —see also Section 20.22.

The ‘M’ notation

This variant of the above ‘middle’ slice notation (e.g., $MR \equiv Rm$) is part of the ‘superset’ notation of Randelshofer. As before, the rotation direction follows that of the designated face. Each has a complementary ‘prime’ version formed by appending a ‘p’. The $M$ must be in upper case.

The MES notation

An alternative but very confusing inner-slice notation (e.g., $Ep \equiv Um$) which is occasionally used is the so-called MES notation as used in the Waterman algorithm (Treep and Waterman 1987), and the Roux method (Giles Roux).

\begin{itemize}
    \item $M$ (\texttt{M}, \texttt{[M]}, \texttt{middle}, between the LEFT and RIGHT faces; direction follows \texttt{L}),
    \item \texttt{E} (\texttt{EQUATOR}, \texttt{[E]}, \texttt{equator}, between the UP and DOWN faces; direction follows \texttt{D}),
    \item \texttt{S} (\texttt{STANDING}, \texttt{[S]}, \texttt{standing}, between the FRONT and BACK faces; direction follows \texttt{F}).
\end{itemize}

Each of these also has an inverse (prime) version.
The ‘S’ notation

Su In this equally confusing inner-slice notation, ‘S’ stands for ‘inner-slice’; the face letter must be in lower case (e.g., Sr \equiv Rm). For example, the inner-slice rotation between the RIGHT and LEFT faces whose rotation direction follows the rotation R is denoted as Sr, which is typeset using the command \rr{Sr}. Each has an inverse (prime) p-form.

Sb

9.4 Outer-slice rotations

The ‘s’ (slice) notation

Us This is a ‘paired’ form of notation (two rotations at once), which can be thought of as complementing the inner-slice (middle layer) rotations. Each of these ‘slice’ commands denotes a rotation of two opposite faces in the same direction.

Ds For example, Us [u] \equiv U \, u + D \cdot u'; i.e., both face-rotations are in the same direction as U. Each of these rotation-codes has a complementary ‘anti-slice’ version (see below).

Fs

Rs This notation was originally described by Singmaster (Frey and Singmaster, 1982), and is much used on the ‘Pretty patterns’ page of the Fridrich website (this page also has a useful link to ‘notation’).

Su This variant of the above ‘slice’ notation (e.g., SU \equiv Us) is part of the ‘superset’ notation of Randelshofer. As before, the rotation direction follows that of the designated FACE. Each has a complementary ‘prime’ version formed by appending a ‘p’.

Sf

Sr

Sb

The ‘a’ (anti-slice) notation

Ua Each of these commands denotes a rotation of two opposite faces in opposite directions. For example, Ua [u] \equiv U \, u + D \, u. This notation is much used on the ‘Pretty patterns’ page of the Fridrich website (see the note above re: ‘slice notation’).

Da

La

Ra

Fa

9.5 Wide rotations

The ‘w’ notation

Uw The clockwise combined rotation of an outer face AND its adjacent inner-slice (officially known as a ‘double block’, or ‘double outer slice’ move) is denoted by appending a lower-case w (denoting ‘wide’) to a rotation-code (endorsed by the WCA). For example, a RIGHT double outer slice rotation \rrh{Rw} is denoted as Rw \rr{Rw}. The ‘prime’ version is formed by appending a ‘p’ to the rotation-code. For example, Rw’ is generated by \rrp{Rwp}.
The ‘T’ notation

This confusing variant of the above ‘w’ notation (e.g., TR ≡ Rw) is part of the
‘superset’ notation of Randelshofer. As before, the rotation direction follows that
of the designated FACE. Each has a complementary ‘prime’ version formed by
appending a ‘p’.

9.6 Axis rotations

The x, y, z notation

Whole-cube clockwise rotations of 90-degrees about about the orthogonal axes
centred on the RIGHT, UP, FRONT faces are denoted as x, y, z (the \rubik{} forms) re-
spectively (see Figure 2), with their hieroglyphs (the \rubikh{} forms) being denoted
as [x], [y], [z] in order to distinguish them from square layer-rotation hieroglyphs.
Note that since x, y, z rotations are always expressed in lower case; this practice
is also extended to the commands.

For example, an x2 rotation (two x rotations one after the other, i.e., [x] [x])
denotes rotating the cube 180 degrees about its x axis so as to bring the DOWN
face into the UP position.

An appended prime ‘ indicates an anticlockwise rotation; for example, x’ (which
is generated by appending a ‘p’ to the rotation-code, i.e., \rubik{xp}).

The \rubik{} forms (and their prime ‘p’ versions) generate the same hiero-
glyphs as their \rubikh{} versions, except that their spacing is similar to that associ-
ated with the ‘square box’ \rubik{} hieroglyphs. Consequently when typesetting
an axis command in a sequence of ‘square-box’ \rubik{} commands, it is better to
use the \rubik{} form rather than the equivalent \rubikh{} form (see the examples
in Section 9.7). There are no \textRubik{} forms for the axis commands (since
they are not necessary).

The u, d, l, r, f, b notation

These are a commonly used alternative for the x, y, z notation (and also endorsed
by the WCA), and denote a 90 degree whole-cube rotation in the same directional
sense as that of the associated face rotation.

Thus d ≡ u’ ≡ y’ etc. For example, d and d’ are generated by the commands
\rubik{d} and \rubikh{dp} respectively. Note that u is the opposite of d, l is the opposite
of r, and f is the opposite of b, etc.

As with the [x], [y], [z] forms (described above) there are also equivalent \rubikh{} and
\rubik{} forms. For example, [d] is generated by the command \rubik{d}.
The ‘c’ notation

\( \text{Uc} \) This slightly more intuitive notation (the ‘c’ stands for ‘cube’) also associates the rotation direction with that of the designated face (e.g., \( \text{Rc} \equiv x \)). Each has a complementary ‘prime’ version formed by appending a ‘p’. For example, \( \text{Rc} \) is equivalent to \( x \); \( \text{Rc}' \) (\( \text{Rcp} \)) is equivalent to \( x' \).

\( \text{Fc} \) This notation, which was probably invented by Singmaster, was originally used on the Cube Lovers usenet group (1981–1997). It is now much used on the Jaap Puzzles website (see Scherphius J) —see also Section 20.22.

The ‘C’ notation

\( \text{CU} \) This variant of the whole cube rotation notation (e.g., \( \text{CR} \equiv \text{Rc} \equiv x \)) is part of the ‘superset’ notation of Randelshofer. As before, the rotation direction follows that of the designated face. Each has a complementary ‘prime’ version formed by appending a ‘p’.

\section{9.7 Examples}

\( \text{R} \) is generated by the command \( \text{\textbackslash rr\{R\}} \)
\( \text{Fw} \) is generated by the command \( \text{\textbackslash rr\{Fw\}} \)
\( \text{L}^2 \) is generated by \( \text{\textbackslash rr\{L\}^2} \)
\( \text{L'2} \) is generated by \( \text{\textbackslash rr\{L\}2} \)
\( \text{Fw}' \) is generated by \( \text{\textbackslash rr\{Fwp\}} \)
\( x \) and \( y \) and \( z' \) are generated by \( \text{\textbackslash rr\{x\}} \) and \( \text{\textbackslash rr\{y\}} \) and \( \text{\textbackslash Rubik\{zp\}} \)
\( \text{Fc} \) and \( \text{Bc} \) are generated by \( \text{\textbackslash rr\{Fc\}} \) and \( \text{\textbackslash rr\{Bc\}} \)
\( \text{UU RR} \) is generated by \( \text{\textbackslash rr\{U\}}\text{\textbackslash rr\{U\}}\text{\textbackslash rr\{R\}}\text{\textbackslash rr\{R\}} \)

\( \text{\Rubik{F}\Rubik{U}\Rubik{y}\Rubik{Rp}\Rubik{Lwp}} \)

\( \text{\text{F } \text{U } \text{R' } \text{Lw'}} \)

Commas can be important in avoiding ambiguity; for example,

\( \text{D, U2, F2, Ds2, B,} \text{\textbackslash rr\{U\}2, \text{\textbackslash rr\{F\}2, \text{\textbackslash rr\{Ds\}2, \text{\textbackslash rr\{B\}, \text{\textbackslash rrh\{U\}2, \text{\textbackslash rrh\{F\}2, \text{\textbackslash rrh\{Ds\}2 \}} \}

\) Finally, if each rotation element uses the \textit{same} font or encoding, for example

\( \text{\Rubik{F}\Rubik{U}\Rubik{y}\Rubik{Rp}\Rubik{Lwp}} \)

then typesetting such a rotation sequence can be achieved more easily using the \texttt{\textbackslash ShowSequence} command (see Section 11). For example, we can typeset the last sequence much more conveniently, as follows:

\( \text{\textbackslash ShowSequence}\{}\text{\textbackslash rr\{F, U, y, Rp, Lwp}\} \rightarrow \text{\Rubik{F}\Rubik{U}\Rubik{y}\Rubik{Rp}\Rubik{Lwp}} \)
9.8 Backwards compatibility

Note that in keeping with ‘backwards compatibility’ all rotation commands (see below) can still be written without the usual curly braces {}. For example, the hieroglyph \[
\text{\texttt{D}}
\] can also be generated using the command \texttt{\texttt{D}}.

9.9 Listing of all rotation commands

Note that all the commands presented here also have a \texttt{\texttt{D}} equivalent form which typesets both the hieroglyph and its lettercode in a vertical format, as shown in the ‘Examples’ section above. These have been omitted here owing to the difficulty of including this form easily in the following table.

Note also that some \texttt{\texttt{D}} commands (e.g., the \texttt{D} command) show only the lettercode in a square box, e.g., \texttt{D}. This is because these rotations do not have a ‘true’ visual representation as seen from the FRONT face, and hence can be somewhat ambiguous unless typeset with their associated lettercode.

<table>
<thead>
<tr>
<th>Command</th>
<th>Hieroglyph</th>
<th>Lettercode</th>
</tr>
</thead>
</table>
| U \texttt{U} | \[
\text{U}
\] | \texttt{\texttt{U}} | \texttt{\texttt{U}} |
| U' \texttt{Up} | \[
\text{U}
\] | \texttt{\texttt{U}} | \texttt{\texttt{Up}} |
| Uw \texttt{Uw} | \[
\text{U}
\] | \texttt{\texttt{U}} | \texttt{\texttt{Uw}} |
| Uw' \texttt{Uwp} | \[
\text{U}
\] | \texttt{\texttt{U}} | \texttt{\texttt{Uwp}} |
| Us \texttt{Us} | \[
\text{U}
\] | \texttt{\texttt{U}} | \texttt{\texttt{Us}} |
| Us' \texttt{Usp} | \[
\text{U}
\] | \texttt{\texttt{U}} | \texttt{\texttt{Usp}} |
| Ua \texttt{Ua} | \[
\text{U}
\] | \texttt{\texttt{U}} | \texttt{\texttt{Ua}} |
| Ua' \texttt{Uap} | \[
\text{U}
\] | \texttt{\texttt{U}} | \texttt{\texttt{Uap}} |
| Um \texttt{Um} | \[
\text{U}
\] | \texttt{\texttt{U}} | \texttt{\texttt{Um}} |
| Um' \texttt{Ump} | \[
\text{U}
\] | \texttt{\texttt{U}} | \texttt{\texttt{Ump}} |
| Uc \texttt{Uc} | \[
\text{U}
\] | \texttt{\texttt{U}} | \texttt{\texttt{Uc}} |
| Uc' \texttt{Ucp} | \[
\text{U}
\] | \texttt{\texttt{U}} | \texttt{\texttt{Ucp}} |
| D \texttt{D} | \[
\text{D}
\] | \texttt{\texttt{D}} | \texttt{\texttt{D}} |
| D' \texttt{Dp} | \[
\text{D}
\] | \texttt{\texttt{D}} | \texttt{\texttt{Dp}} |
| Dw \texttt{Dw} | \[
\text{D}
\] | \texttt{\texttt{D}} | \texttt{\texttt{Dw}} |
9.9.1 Randelshofer notation

\begin{align*}
\text{CR} & \quad \text{CR} \quad \text{CR} \quad \text{CR} \\
\text{CR'} & \quad \text{CR'} \quad \text{CR'} \quad \text{CR'}
\end{align*}
\textRubik{SL'} \rr{SLp} \textRubik{SL'} \rrh{SLp} \textRubik{SL'} \rrh{SLp} \\
\textRubik{SU} \rr{SU} \textRubik{SU} \rrh{SU} \textRubik{SU} \rrh{SU} \\
\textRubik{SU'} \rr{SUP} \textRubik{SU'} \rrh{SUP} \textRubik{SU'} \rrh{SUP} \\
\textRubik{SD} \rr{SD} \textRubik{SD} \rrh{SD} \textRubik{SD} \rrh{SD} \\
\textRubik{SD'} \rr{SDp} \textRubik{SD'} \rrh{SDp} \textRubik{SD'} \rrh{SDp} \\
\textRubik{SF} \rr{SF} \textRubik{SF} \rrh{SF} \textRubik{SF} \rrh{SF} \\
\textRubik{SF'} \rr{SFp} \textRubik{SF'} \rrh{SFp} \textRubik{SF'} \rrh{SFp} \\
\textRubik{SB} \rr{SB} \textRubik{SB} \rrh{SB} \textRubik{SB} \rrh{SB} \\
\textRubik{SB'} \rr{SBp} \textRubik{SB'} \rrh{SBp} \textRubik{SB'} \rrh{SBp} \\
\textRubik{TR} \rr{TR} \textRubik{TR} \rrh{TR} \textRubik{TR} \rrh{TR} \\
\textRubik{TR'} \rr{TRp} \textRubik{TR'} \rrh{TRp} \textRubik{TR'} \rrh{TRp} \\
\textRubik{TL} \rr{TL} \textRubik{TL} \rrh{TL} \textRubik{TL} \rrh{TL} \\
\textRubik{TL'} \rr{TLp} \textRubik{TL'} \rrh{TLp} \textRubik{TL'} \rrh{TLp} \\
\textRubik{TU} \rr{TU} \textRubik{TU} \rrh{TU} \textRubik{TU} \rrh{TU} \\
\textRubik{TU'} \rr{TUp} \textRubik{TU'} \rrh{TUp} \textRubik{TU'} \rrh{TUp} \\
\textRubik{TD} \rr{TD} \textRubik{TD} \rrh{TD} \textRubik{TD} \rrh{TD} \\
\textRubik{TD'} \rr{TDP} \textRubik{TD'} \rrh{TDP} \textRubik{TD'} \rrh{TDP} \\
\textRubik{TF} \rr{TF} \textRubik{TF} \rrh{TF} \textRubik{TF} \rrh{TF} \\
\textRubik{TF'} \rr{TFP} \textRubik{TF'} \rrh{TFP} \textRubik{TF'} \rrh{TFP} \\
\textRubik{TB} \rr{TB} \textRubik{TB} \rrh{TB} \textRubik{TB} \rrh{TB} \\
\textRubik{TB'} \rr{TBp} \textRubik{TB'} \rrh{TBp} \textRubik{TB'} \rrh{TBp} \\

9.10 The rubikfont

For hieroglyph-related text we use the standard Computer Modern Sans (cmss) bold extended (bx) 10pt font for upper-case letters, and the 8pt footnote size for lower-case letters (see Section 20.3 for details). This font (rubikfont) and the ‘prime’ symbol (rubikprime) can be easily changed by ‘renewing’ the three
commands there.

For example, to change to the somewhat ‘lighter’ semi-bold extended (sbx) CM Sans (cmss) form one can simply include the following in the preamble (the FNS suffix stands for ‘footnotesize’):

\makeatletter
\renewcommand{\rubikfont}{\fontsize{10}{12pt}\usefont{T1}{cmss}{sbx}{n}}
\renewcommand{\rubikfontFNS}{\fontsize{8}{12pt}\usefont{T1}{cmss}{sbx}{n}}
\makeatother

The ‘rubikprime’ symbol

We currently use the apostrophe for the prime symbol (see Section 20.3), since the maths \texttt{\prime} seems to be a bit too faint (especially since we need to use the ‘scriptstyle’ size in this setting). However, users can easily make the Rubik bundle use the maths prime instead, by loading the following in the preamble.

\makeatletter
\renewcommand{\rubikprime}{\raisebox{1.2pt}{\ensuremath{\scriptstyle{\prime}}}}
\makeatother

10 Draw commands

A \texttt{\Draw..} command typesets either a cubie, cube or face using parameters set or defined via previous parameter-allocation commands (e.g., colours, dimensions etc).

It is important to distinguish between the \texttt{\rubikcube} package \texttt{\Draw..} commands (with an upper-case D) and TikZ \texttt{\draw..} commands (with a lowercase d). Rubik \texttt{\Draw..} commands are implemented by the TikZ \texttt{\draw..} commands, and consequently \texttt{\Draw..} commands can only be used inside a TikZ picture environment—and hence they can also be used safely in conjunction with the \texttt{\ShowCube} command, which itself uses a TikZ picture environment. See also Section 10.1 below.

There are six types of \texttt{\Draw..} commands, as follows:

\texttt{\DrawCubie..}
\texttt{\DrawRubikCube..}
\texttt{\DrawRubikCubeSidebar..}
\texttt{\DrawRubikFace..}
\texttt{\DrawRubikFlat..}
\texttt{\DrawNCube..}

Note that the former \texttt{\DrawRubikLayer..}, \texttt{\DrawCube..}, \texttt{\DrawFace..} commands are now deprecated, since they have been superseded by more versatile and intuitive commands (see Section 15).
10.1 \texttt{\textbackslash RubikBundle} error message

If a \texttt{\textbackslash RubikBundle} command is used outside a TikZ picture environment, then \LaTeX{} issues an "Undefined control sequence" error message, indicating that it is trying to draw something using an undefined TikZ \texttt{\textbackslash draw} command\footnote{Note that the TikZ \texttt{\textbackslash draw} command uses a lower-case 'd', while all \texttt{\textbackslash RubikBundle} commands start with an upper-case letter.}

This is because all Rubik \texttt{\textbackslash RubikBundle} commands achieve their effects by implementing a series of TikZ \texttt{\textbackslash draw} and other commands, all of which need to be inside a \texttt{tikzpicture} environment.

For example, if the command \texttt{\textbackslash RubikBundleCubeF} is used without a surrounding TikZ picture environment, then something similar to the following error message will be generated.

! Undefined control sequence.\texttt{\textbackslash RubikBundleCubeF}!

10.2 \texttt{\textbackslash RubikBundleCube} commands

\texttt{\textbackslash RubikBundleCubeXY} This command draws a single cubie in one of four orientations as denoted by the terminal XY viewing-direction codes. Since a single cubie has only three visible faces this command takes three xyz-ordered colour parameter arguments. Consequently the \texttt{\textbackslash RubikBundleCube} command has the format

\texttt{\textbackslash RubikBundleCubeXY\{x\}\{y\}\{z\}}

where the XY pair denotes the viewing direction as before, and the xyz parameters denote the face colours associated with each of the three axes.

For example, the command \texttt{\textbackslash RubikBundleCubeRU\{O\}\{Y\}\{G\}} draws a single cubie as viewed from the RightUp direction, with face colours Orange (x-axis), Yellow (y-axis), Green (z-axis), as follows.

\begin{figure}
\centering
\includegraphics[width=0.2\textwidth]{example}
\caption{Example of a single cubie drawn from the RightUp direction.}
\end{figure}

\texttt{\textbackslash ShowCube\{1.33cm\}\{1\}\{\textbackslash RubikBundleCubeRU\{O\}\{Y\}\{G\}\}}

Since the front face is 1 unit wide and the 2D width of the side approx 1/3 unit, and the scale-factor = 1, then the minipage width required for the cubie image = (1.33 × 1) = 1.33cm.

\texttt{\textbackslash RubikBundleCubiedy} Minor cubie configuration changes can be effected by adjusting the Cubiedy and Cubiedx values (> 0; no units) shown in Figure 3 via the two commands

\texttt{\textbackslash RubikBundleCubiedy\{\}}
\texttt{\textbackslash RubikBundleCubiedx\{\}}

as shown in the following example.
For convenience, there are also four (smaller) ‘text’ versions of the four \textCubie commands for use in ordinary text, as follows:

\textCubieRU{O}{Y}{G}
\textCubieRD{O}{Y}{G}
\textCubieLU{O}{Y}{G}
\textCubieLD{O}{Y}{G}

Note that these \textCubie commands are not influenced by the \Cubiedy, \Cubiedx commands as their size is pre-set for text use.

10.4 DrawRubikCube commands

This command draws Rubik cubes in one of four oblique orientations or configurations as denoted by the following terminal \textX viewing-direction codes: RU (RightUp), RD (RightDown), LU (LeftUp), LD (LeftDown); two additional terminal codes are F (Flat) and SF (Semi-Flat). For example, the command

\DrawRubikCube{1.7cm}{1}%
\Cubiedy{0.4}
\Cubiedx{0.8}
\DrawCubieRU{O}{Y}{G}
\begin{verbatim}
\DrawRubikCubeRU
will draw a Rubik cube as viewed from the RightUp direction (RU), as shown in
the following figure.
\end{verbatim}

\begin{verbatim}
\DrawRubikCubeF
This command draws the completely flat (F) format of the cube, as shown in
the following example.
\end{verbatim}

The addition of text (numbers or letters) in the faces is straightforward—the
origin of the 1-unit grid is located at the bottom left corner of the front
face (orange here). The letters were placed using the following TikZ code inside the
TikZ picture environment (remember TikZ commands require a terminal semi-
colon ;).

\begin{verbatim}
\DrawRubikCubeSolved
\ShowCube{5cm}{0.4}{\DrawRubikCubeF}
\end{verbatim}

\begin{verbatim}
A useful ‘semi-flat’ (SF) alternative format, which uses the standard RU view
of the cube and appends the three hidden sides (cf., Rokicki et al., 2013), is
generated by the command \DrawRubikCubeSF as follows.
\end{verbatim}
Note that even in this configuration it is straightforward to write text on the graphic, since the 2D width (on the page) of the green right face is exactly 1-unit, and the bottom right-hand corner of the green face is raised exactly 1-unit (see Figure 1). Consequently, since the origin of the coordinate-grid is at the bottom left corner of the front face (the orange face here), the \((x, y)\) coordinates of the centre of the red back face are easily determined to be \((5.5, 2.5)\).

10.5 \texttt{DrawRubikFace}.. commands

These commands draw the current state of a specified face (e.g., \texttt{DrawRubikFaceUp}), or the face and all the associated sidebars (e.g., \texttt{DrawRubikFaceUpSide}). These commands do not take any arguments—for code see Section 20.15.

Note: These commands replace the earlier \texttt{DrawFace}.. commands (see Section 15).

For example, a simple way to show the yellow-cross configuration in the up face would be to first define the colours using the \texttt{RubikFaceUp} command, and then draw the up face using the \texttt{DrawRubikFaceUp} command, as follows:

\begin{verbatim}
\RubikFaceUp{x}{y}{x}
\ShowCube{2.1cm}{0.7}{\DrawRubikFaceUp}
\end{verbatim}

10.6 Sidebars & \texttt{DrawRubikFaceXSide} commands

In the next example we use the \texttt{DrawRubikFaceUpSide} command to draw the up face and all its sidebars in a cube having a ‘solved’ WY (White opposite Yellow) configuration.

\begin{verbatim}
\RubikCubeSolvedWY
\ShowCube{1.6cm}{0.5}{\DrawRubikFaceUpSide}
\end{verbatim}

Short-hand versions: For convenience each of these commands has an equivalent short-hand version generated by using just the first letter of the face name and
(where appropriate) the first letter of the word Side. For example, \\
\DrawRubikFaceR ≡ \DrawRubikFaceRight,  \\
\DrawRubikFaceRS ≡ \DrawRubikFaceRightSide, etc.

10.7 Sidebar parameters

The default values (size) of the sidebars are as follows: width (0.3), length(1) and separation from the square face (0.3) —see Section 20.16 for the code. Note that the default value of the length of a cubie side is 1. These sidebar values (decimal values ≥ 0; no units) can be changed from their default values using the three commands. 

\RubikSidebarWidth{} (default = 0.3) \\
\RubikSidebarLength{} (default = 1.0) \\
\RubikSidebarSep{} (default = 0.3) 

Values set in the document preamble will apply globally, while values set within a TikZ picture environment will apply only locally to that particular environment. Alternatively, one can keep the effect local using braces (see below).

In the following example, we show the effect on the up face and sidebars of a normally solved (WY) cube after dramatically changing the sidebar width, length and separation from the default values—compare with the previous image. For convenience, we have used a pair of braces to keep the effect local to this example.

\begin{verbatim}
{ \\
\RubikCubeSolvedWY \\
\RubikSidebarWidth{0.8} \\
\RubikSidebarLength{0.5} \\
\RubikSidebarSep{0.7} \\
\ShowCube{2cm}{0.5}{\DrawRubikFaceUpSide}
}
\end{verbatim}

Note also that changing the sidebar-width or sidebar-separation values may well also change the surrounding white-space (use \fbox to visualise this) and may therefore require some fine-tuning of the minipage width setting in order to optimise appearance.

10.8 \NoSidebar command

The \NoSidebar{(colour-code)} command (which takes a single colour code argument) allows the user to disable the drawing of sidebars having a particular colour (for code see Section 20.16.1). Its action can be localised by placing the command inside an environment (e.g., inside the \ShowCube environment). Alternatively, the action of this command can be disabled simply by writing it with an empty argument, e.g., \NoSidebar{.}.
This command is designed to facilitate the drawing of so-called OLL (Orientate Last Layer) configurations, which are typically rendered using the yellow face.

For example, the following figure uses the \DrawRubikFaceUpSide command to draw the commonly encountered OLL configuration known as the ‘yellow cross’ (the remaining four yellow facelets associated with this layer are shown as sidebars). In this example, we first define the colours for the whole cube (grey), and then redefine the colours for the up face and its four adjacent faces. Finally we draw the up face and sidebars; we also show an alternative way of writing the facelet colour codes (ie without using the curly brackets).

\RubikCubeGreyAll
\RubikFaceUp YXY YYY YXY
\RubikFaceFront YXY XXXXX
\RubikFaceRight XXY XXXXX
\RubikFaceBack XXX XXXXX
\RubikFaceLeft YXX XXXXX
\ShowCube{2.6cm}{0.6}{\DrawRubikFaceUpSide}

However, we can greatly improve the OLL image by disabling the drawing of all the grey (X) sidebars by using the \NoSidebar{X} command as follows (here we have placed the \NoSidebar{X} command inside the \ShowCube environment in order to limit its action locally). Note also that this time we have used the short-hand US (UpSide) version of the \DrawRubikFaceUpSide command.

\RubikCubeGreyAll...
\ShowCube{2.6cm}{0.6}{\NoSidebar{X}\DrawRubikFaceUS}

10.9 Cube sidebars & DrawRubikCubeSidebar commands

Cube sidebars are drawn adjacent to cube edges which are defined by the two faces forming the edge. Thus the BR (Back-Right) sidebar is placed adjacent to the edge formed by the BACK and RIGHT faces.

Since the cube orientation (view direction: RU, RD, LU, LD) determines which sidebars are visible, the command for drawing the sidebar (DrawRubikCubeSidebar...) also needs to incorporate the view direction. The command takes two mandatory arguments: the first is the pair of face codes defining the edge (XX); the second is the view direction, as follows:

\DrawRubikCubeSidebarXX{{\em view direction}}

EXAMPLE: \DrawRubikCubeSidebarRB{LD}
Note that the pair of face codes are order independent, and hence can be written in any order, which makes remembering the commands very easy.

Note also that at present commands are only available for the eight sidebars which are parallel to X,Y axes, as these seem to be the most useful.

In the following example, we input a previously saved cube state (in the file \texttt{cubestate-A.tex}) and draw the cube and the four main sidebars (BR, BD, FL, FU) visible from the RD view direction.

```
\input{cubestate-A.tex}
\ShowCube{3cm}{0.6}{
  \DrawRubikCubeRD
  \DrawRubikCubeSidebarBR{RD}
  \DrawRubikCubeSidebarBD{RD}
  \DrawRubikCubeSidebarFL{RD}
  \DrawRubikCubeSidebarFU{RD}
}
```

10.10 \texttt{DrawRubikFlat} commands

The \texttt{DrawRubikFlat\{\langle x\rangle\}{\langle y\rangle}} commands draw a ‘flat’ square representation of a specified face, located such that its bottom left corner is positioned at \((x,y)\).

Each command (except \texttt{DrawRubikFlatFront}) takes two arguments, namely the X-coordinate and Y-coordinate of the bottom left corner of the face. This \((x,y)\) pair allows the user to position the face (see Section 20.14 for the code).

These commands are designed to supplement the \texttt{DrawRubikCube...} commands and allow hidden faces to be represented.

Note also that the \texttt{DrawRubikFlatFront} command currently takes no arguments, since by definition the bottom left corner of this face is always at \((0,0)\), and there seems to be no reason (just now) for this face to have the \((x,y)\) facility.

Usage: The following example uses the command \texttt{DrawRubikFlatBack\{4\}{1}} to append the back face to the side of a 3D cube. Note that since the coordinates of the bottom/back/right corner of the cube rendered by the command \texttt{DrawRubikCubeRU} is \((4,1)\) (see Section 4), we can position the lower/left corner of the back face at this point using the command \texttt{DrawRubikFlatBack\{4\}{1}} as follows:

```
\RubikCubeSolvedWY
\ShowCube{3cm}{0.5}{%
  \DrawRubikCubeRU
  \DrawRubikFlatBack\{4\}{1}
}
```

10.11 \texttt{DrawNCube} (NxNxN)

An ‘NCube’ is a solved NxNxN cube drawn from the RU direction: (i.e., only shows faces UP, FRONT, RIGHT). The cubic colours of each face are All the same.
\DrawNCubeAll{N}{Xcolour}{Ycolour}{Zcolour}.

This command takes four ordered parameters (N, X, Y, Z)—the number (integer; N > 0) of cubies along an edge, followed by three face colours in XYZ order. Since the viewpoint is only from the RU direction, the three colour parameters are: X(Right), Y(Up), Z(Front).

\ShowCube{3.5cm}{0.5}{\DrawNCubeAll{5}{O}{Y}{G}}

11 \ShowSequence command

The \ShowSequence{(separator)}{(font-code)}{(sequence)} command typesets a comma-separated sequence of rotation codes—for code see Section 20.5. Appending an ‘F’ to the command-name (\ShowSequenceF) results in an fbox around the whole output (cf. the \ShowCubeF command). Appending a lower-case ‘f’ to the command-name (\ShowSequencef) generates fboxs around each element in the output (these two forms can be helpful when checking white space).

The \ShowSequence command takes three mandatory arguments: the first is the separator (#1), the second is the font or style code (#2), and third is a comma-separated sequence of Rubik rotation codes (#3).

The separator (e.g., comma) used for typesetting the sequence can be specified (or just omitted—i.e., an empty bracket); the sequence can be either a named sequence (i.e., encoded as a macro) or just a comma separated sequence of rotation codes.

Usage: The following command displays the rotation sequence F, R, U, Rp, Up, using the \rr encoding and comma & space separated.

\ShowSequence{,\ }{\rr}{F,R,U,Rp,Up} \rightarrow F, R, U, R', U'

Remember that if you want a very long sequence to break automatically at the line-ends, then you need to include at least one space between the sequence elements, either with or without a comma; for this the separator argument needs to include a space, i.e., something like {, \ } or { \ } or just { } perhaps.

For example, the following commands typeset the comma-separated rotation sequence U,D,Lp,R using different separators and fonts. For convenience, we start by encoding the sequence as the macro \myseq.

\newcommand{\myseq}{U,D,Lp,R} \ShowSequence{,}{\texttt}{\myseq} \rightarrow U,D,Lp,R
\ShowSequence{,}{\rr}{\myseq} \rightarrow \text{U,D,L',R}
\ShowSequence{}{\rrh}{\myseq} \rightarrow \text{[seqA],L,R,D2,L3,<test>}
\ShowSequence{}{\textRubik}{\myseq} \rightarrow \text{[seqA],L,R,D,D,L,L,L}
\ShowSequence{}{\texttt}{\myseq} \rightarrow \text{U,E,D,E,L',R,E}

11.1 Trailing digits

Note that the commands \rr, \rrh, \Rubik and \textRubik will fail for rotation codes which have a trailing digit, e.g., R2, Dp3 (so-called ‘short’ codes), since the \ShowSequence macro currently only reads the whole string between pairs of commas (i.e., it does not interpret strings like R2 \rightarrow R,R, etc.). However, if you just want to ‘see’ the text, then the standard LaTeX typewriter command \texttt will typeset the text correctly, as you would expect (but using the \tt font of course).

A useful ‘work-around’ for this limitation is provided by the \RubikRotation{} command \textsuperscript{11} (part of the \rubikrotation package), since this returns (via the Perl program \rubikrotation.pl) an extended (‘long’) form of a given sequence in which any ‘short’ rotation codes are converted into their separate (atomic) long-form rotation codes; the associated string is \SequenceLong. In fact three different forms of the sequence are returned, as shown below.

For example, consider the following short-form sequence L,R,D2,L3, and denote it as ‘seqA’ as follows: \newcommand{\seqA}{[seqA],L,R,D2,L3,<test>}. In this case D2,L3, and the name ‘seqA’ fail to be typeset when using the \rr font with the \ShowSequence command (notice there are lots of commas with nothing between them), while \texttt does work, as follows:
\newcommand{\seqA}{[seqA],L,R,D2,L3,<test>}
\ShowSequence{}{\rr}{\seqA} \rightarrow \text{[,L,R,,}
\ShowSequence{}{\texttt}{\seqA} \rightarrow \text{[seqA],L,R,D2,L3,<test>}

However, if we now run the command \RubikRotation{\seqA} we shall then have at our disposal the following four strings:
\SequenceInfo = test
\SequenceName = seqA
\SequenceShort = L,R,D2,L3
\SequenceLong = L,R,D,D,L,L,L

and therefore have full control over typesetting rotation sequences. This approach therefore offers sufficient flexibility for most purposes. For example, if we now write the command \ShowSequence{}{\rr}{\SequenceLong} we obtain
\ShowSequence{}{\rr}{\SequenceLong} \rightarrow \text{L,R,D,D,L,L,L}

In practice, any trailing digit is converted (modulo 4) and expanded accordingly, the details being copied to the log-file. For example, if one writes the

\textsuperscript{11}Using the \texttt{--shell-escape} command-line option with the \LaTeX engine—see Section 1.2.
command \RubikRotation{\texttt{test}}, then since $28978 \equiv 2 \pmod{4}$ then \SequenceLong will be returned containing the string $R,D,D$. The associated entries in the log-file are as follows:

```latex
...rotation R OK
...Expanding D28978 (28978 = 2 mod 4) ...
...rotation D OK (\texttt{Dp3})
...rotation D OK (\texttt{Dp3})
...writing new Rubik state to file rubikstateNEW.dat
...SequenceName = \texttt{test}
...SequenceShort = R,D28978
...SequenceLong = R,D,D
```

### 12 SequenceBrace commands

The \SequenceBraceX{\texttt{name}}\{\texttt{sequence}\} command is a tool for displaying a named sequence using a brace. The trailing A denotes that the brace is placed Above the sequence; the trailing B denotes the brace is Below the sequence. Appending an ‘F’ to the command-name (e.g., \SequenceBraceAF) results in a surrounding fbox (cf. the \ShowCubeF command). For the code see Section 20.6.

A typical example of its use might be:

```latex
\newcommand{\fourspot}{\{\texttt{fourspot}\},F2,B2,U,Dp,R2,L2,U,Dp}\
\noindent\
\RubikCubeSolvedWY\
\ShowCube{1.6cm}{0.4}{\DrawRubikCubeRU}\
\RubikRotation{\fourspot}\
\quad\SequenceBraceA{\SequenceName}{\texttt{\ShowSequence}}{\Rubik}{\SequenceLong}\
\quad\longrightarrow\
\ShowCube{1.6cm}{0.4}{\DrawRubikCubeRU}\
```

Note (1) that the ‘fourspot’ listing is included in the RUBIKPATTERNS package, and (2) the \RubikRotation command requires using the \texttt{--shell-escape} command-line option with the \TeX{} engine—see Section 1.2 for details.
13 Arrows

The \rubikcube package does not offer any special commands for drawing arrows since it is straightforward just to include the appropriate TikZ 'draw' commands in the \tikzpicture environment (i.e., in our own \ShowCube environment).

In order to facilitate using the standard TikZ 'draw' commands the graphic grid origin of Rubik cube images is located at the bottom left corner of the FRONT face (see also Section 4 for details of the cube’s coordinate system). Similarly, single face images (e.g., drawn using the \DrawRubikFace.. or \DrawRubikFlat.. commands etc.) and also Sidebars, have their grid origin at the bottom left corner of the face. Consequently the start and finish coordinates for any arrow or line are easy to determine.

![Figure 4: Grid showing the positive coordinates associated with FRONT face of a cube image, or of a face-only image.](image)

For example, Figure 4 shows a green arrow drawn from the centre of the lower-left cubie (0.5, 0.5) to the centre of middle cubie (1.5, 1.5). To do this we just include the following TikZ command (remember that TikZ commands require a terminal semicolon):

\begin{verbatim}
\draw[->,color=green] (0.5,0.5) -- (1.5, 1.5);
\end{verbatim}

The following example shows the cubie movement in the UP face generated by the rotation sequence FRUR’U’F’. The magenta arrows indicate movement with cubie rotations, while the black arrow indicates movement without rotation. This example also highlights the fact that when there are several arrows, the start and end positions often need to be offset slightly away from cubie centres.

\begin{verbatim}
\RubikCubeSolvedWY
\ShowCube(2.5cm){0.7}{% 
\DrawRubikFaceUp
\draw[->,thick,color=magenta] (1.5,0.5) -- (2.4, 1.4);
\draw[->,thick] (2.5,1.5) -- (1.6, 2.4);
\draw[->,thick,color=magenta] (1.3, 2.3) -- (1.3, 0.5);
\draw[<->,thick,color=magenta] (0.5,2.6) -- (2.5, 2.6);
\draw[<-,thick,color=magenta] (0.5,0.3) -- (2.5, 0.3);
}
\end{verbatim}

Since the coordinates shown in Figure 4 extend outwards in all directions, they
can also be used as a guide for drawing arrows (or other structures) outside this 3x3 ‘face’ square. The origin is at lower left corner of the face.

In the following example, we input a Rubik cube configuration (previously saved as the file CubeFour.tex (see the RUBIKROTATION package documentation for details)\textsuperscript{12}, and draw an arrow to highlight a yellow side facelet.

\input{CubeFour.tex}
\ShowCube{2.2cm}{0.5}{\DrawRubikFaceUpSide%
  \draw[->,ultra thick,color=green] (2.5,5) -- (2.5, 4); }

The following example shows an arrow on the Rubik cube. The origin of coordinates is at the bottom left corner of the FRONT face (see Section 4).

\RubikFaceFront{{O}{O}{O}}
{O}{O}{X}
{X}{O}{X}

\RubikFaceRight{{G}{G}{G}}
{X}{G}{G}
{X}{X}{X}

\RubikFaceDown {{X}{G}{X}}
{X}{Y}{X}
{X}{X}{X}

\ShowCube{3cm}{0.7}{%\DrawRubikCubeSidebarFD{RU}
  \draw[ultra thick,->,color=blue]
    (1.5,0.5) -- (2.5, 1.5); }

In the following example we use a blue circle to highlight a corner cubie to be rotated into the top layer. Note we also use the command \DrawRubikCubeSidebarFD{RU} to draw the sidebar along the FD (Front-Down) edge, and show the colour of the hidden facelet of the corner-cubie.

\textsuperscript{12}See also the ‘SaveRubikState’ example in the file RubikExamples.pdf.
14 Final example

We now present, as a final example, the code used to draw the front page figure\textsuperscript{13}. This code uses the \texttt{RubikRotation} command (from the \texttt{rubikrotation} package), and therefore the \LaTeX{} engine needs to be run using the \texttt{--shell-escape} command-line switch (see Section 1.2 for details).

\begin{verbatim}
\noindent\hfil \RubikCubeSolvedWY \ShowCube{1.6cm}{0.4}{\DrawRubikCubeRU} \quad \ShowCube{1.6cm}{0.4}{\DrawRubikFaceUpSide} \hfil

\RubikRotation{F,R,U,Rp,Up,Fp} \ShowSequence{}{\Rubik}{\SequenceLong} \longrightarrow \ShowCube{1.6cm}{0.4}{\DrawRubikFaceUpSide} \hfil
\end{verbatim}

\textsuperscript{13}This is a well-known sequence of order 6 used to cycle three edge cubies; it is used to generate the ‘cross’ configuration in the final layer when solving the cube. Here we are performing the sequence on a ‘solved’ cube, since this allows you to see how the three edge cubies move, and either flip (magenta arrows) or do not flip (black arrow).
14.1 Without using \RubikRotation

If you really need to draw the above figure without using the RUBIKROTATION package (as we had to in order to write this particular document) then you would need to replace the commands

\RubikRotation{F,R,U,Rp,Up,Fp}
\ShowSequence{,}{\Rubik}{\SequenceLong} \quad\longrightarrow\quad \ShowCube{2cm}{0.4}{\DrawRubikFaceUpSide}

with the following commands:

\Rubik{F}\Rubik{R}\Rubik{U}\Rubik{Rp}\Rubik{Up}\Rubik{Fp}
\quad\longrightarrow\quad \RubikFaceUp WWB WWOWRB
\RubikFaceBack RGG XXXXXX
\RubikFaceLeft RBO XXXXXX
\RubikFaceFront GWO XXXXXX
\RubikFaceRight WWW XXXXXX
\quad\ShowCube{1.6cm}{0.4}{\DrawRubikFaceUpSide}

15 Deprecated commands

The \DrawRubikLayerFace.. and \DrawRubikLayerSide.. are now deprecated; they were found to be confusing since (a) they both drew faces and took colour arguments for facelets, and (b) they did not update the internal colour state of a cube or face—i.e., their colouring was simply a local ‘painting’ action without memory. They have both been superseded by the more versatile \DrawRubikFace.. commands (see Section 10.5).

The \RubikSide.. commands are also deprecated, since they duplicated some of the function of the \RubikFace.. commands.

The earlier \DrawFace.. (v4) and \DrawFlat.. (v3) commands are also deprecated since they lacked the Rubik or Two keyword (see Sections 3.1 and 10.5).

Summary of all deprecated commands (and their current versions) since v3.

\DrawRubikCubeFlat \quad\longrightarrow\quad \DrawRubikCubeSF (Semi-Flat)
\DrawRubikFlat \quad\longrightarrow\quad \DrawRubikCubeF (Flat)
\DrawFace.. \quad\longrightarrow\quad \DrawRubikFace..
\DrawFlat.. \quad\longrightarrow\quad \DrawRubikFlat..
\DrawFlat..Side \quad\longrightarrow\quad \DrawRubikFace..Side
\DrawRubikLayerFace.. \quad\longrightarrow\quad \DrawRubikFace..
\DrawRubikLayerSide.. \quad\longrightarrow\quad \DrawRubikFace..

16 Known limitations

Please contact the authors regarding any ideas for improvement, errors, problems or shortcomings etc.
• Note that the rotation hieroglyphs are optimised for a 10pt font and do not scale with document font size. However, they do work well in conjunction with the standard 11pt and 12pt document fonts sizes. Nevertheless, the font size can of course be changed by renewing the font command (see Section 9.10 for details).

• The sidebars cannot be arbitrarily positioned (note: hidden faces can be arbitrarily positioned).

17 Change history

• Version 5 (February 2018)
  — Removed some now unnecessary commands (see Section 15) and made some of the internal code a bit more efficient.
  — Added two new ‘grey’ commands: \RubikCubeGreyWY and \RubikCubeGreyWB (so as to complement the related \RubikCubeSolvedXX commands).
  — Bugfix: added a pair of containing braces around the \ifthenelse.. command used by the \ShowSequence command (for code see Section 20.5). This fixed an occasional problem of a font not being contained.
  — New \NoSidebar command for disabling the drawing of sidebars of a particular colour (see Section 10.8).
  — Implemented the terminal $x, y$ position parameters for the \DrawRubikFlatLeft and \DrawRubikFlatRight commands; all the \DrawRubikFlat.. commands work correctly now (see Section 20.14).

• Version 4.0 (March 2017)
  — Improved documentation.
  — Improved inter-hieroglyph spacing and vertical position. The Computer Modern sans bold font (10/12pt) is used for the hieroglyphs and rotation codes (see Section 20.3 for details).
  — Improved the \showcube and \showcubeF macros (see Sections 6 and 20.4).
  — Additional notation for middle slice rotations (‘m’ notation), e.g., \Rm, \Rm etc (see Sections 9.3 and 20.22).
  — Additional notation for whole cube rotations (‘c’ notation), e.g., \Rc, \Rc etc (see Sections 9.6 and 20.22).
  — Added Randelshofer notation (the ‘CMST’ rotations), e.g., \CR, \MR etc (see Sections 9.9.1 and 20.23).
  — Six new commands for showing and annotating rotation sequences; the versions with a terminal ‘F’ also surround the object with an hbox to allow users to see the extent of any associated white space (see Sections 11 & 12):
— A new command for setting up or allocating a ‘solved’ colour configuration. (see Section 8.3):

\RubikSolvedConfig

— A new command for setting up a ‘starter cube’ for which the whole cube is allocated the default ‘grey’ colour (see Section 8.5):

\RubikCubeGreyAll

— A new supporting RUBIKPATTERNS package has been added to the Rubik bundle. It is a small macro database of well-known named Rubik patterns and associated sequences (see Section 1.4).

• Version 3.0 (September 2015)

— All rotation commands can now use the rotation-code as an argument; for example, the rotation D can now be typeset using the command \rr{D} etc (see Section 9). The new rotation commands are:

\rr{⟨rotation-code⟩}
\rrh{⟨rotation-code⟩}
\Rubik{⟨rotation-code⟩}
\textRubik{⟨rotation-code⟩}

The original rotation command formats (e.g., \rrD) are still supported for backwards compatibility.

— \ShowCube and \ShowCubeF are new commands for displaying a cube inside a minipage (see Sections 6 and 20.4).

— \RubikCubeGrey is a new command for setting up a ‘starter cube’ for which the only allocated colours are those for the centre cubies (see Section 8.5). The colour configuration matches that of the \RubikCubeSolved.

• Version 2.2 (January 2015)

— Fixed typos and minor errors in the documentation.

— Added the following commands to facilitate typesetting a face (but see Section 15).

\DrawFlatUp
\DrawFlatDown
\DrawFlatLeft
\DrawFlatRight
\DrawFlatFront
\DrawFlatBack
\DrawFlatUpSide
\DrawFlatDownSide
\DrawFlatLeftSide
\DrawFlatRightSide
\DrawFlatFrontSide
\DrawFlatBackSide

— Changed ‘Equator’ → ‘Middle’ in all \DrawLayer.. commands (for consistency). Hence ‘E’ → ‘M’ in all Flat commands and Slice commands. Note that although the former use of ‘Equator’ is retained for backward compatibility (for the moment) it is now deprecated.

— Fixed a conflict with the T\TeX \sb command as used by the \url package which resulted in reference chaos when the \url package was used with the R\UBIK\CUBE package (internalised \sb to \@sb). Also internalised, for convenience, \sd to \@sd; \sh to \@sh; \sc to \@sc; \sq to \@sq.

• Version 2.0 (February 5, 2014)
  — First release.

18 Acknowledgements

We thank Peter Bartal and Peter Grill for useful ideas and suggestions; we have built on some of their ideas and have acknowledged these instances in the documentation. We also thank Christian Tellechea for the \@join{}{} command (see Section 20.21.3), Christian Schröppel for help regarding the forarray package (see Section 20.5), Herbert Kociemba for helpful comments, and Robert Mafík for suggesting the \NoSidebar{}{} command (see Section 10.8).

19 References


  http://duvoid.fr/rubik/rubik-debutant-couleurs.pdf  
  http://duvoid.fr/rubik/sources/notation_en.eps  
  http://duvoid.fr/rubik/sources/rubik-debutant-couleurs.tex

  http://duvoid.fr/rubik/sources/rubik-friddrich-couleurs.tex

[re: preventing extra white space]

• Fridrich website (Fridrich J).  
  http://www.ws.binghamton.edu/fridrich/.  
  See the useful ‘notation’ section on the ‘Pretty patterns’ webpage at  
  http://www.ws.binghamton.edu/fridrich/ptrns.html.


• Fung website (Fung A). Solving the Rubik’s cube systematically.  
  http://alexfung.info/favorite/cube/cube.htm


• Golomb SW (1982). Rubik’s cube and quarks: twists on the eight corner cells of Rubik’s cube provide a model for many aspects of quark behaviour.  
  http://www.jstor.org/stable/27851433


• Harris D (2008). Speedsolving the cube. (Sterling Publishing Co. Inc., New York, USA.) pp. 166. [covers 2x2x2, 3x3x3, 4x4x4, 5x5x5 cubes]

• Harris website (Harris D).  
  http://www.cubestation.co.uk

• Heise website (Heise R). Rubik’s cube theory.  
  http://www.ryanheise.com/cube/theory.html

  Scientific American; March issue.

  http://math.berkeley.edu/~hutching/rubik.pdf


• Longridge website (Longridge M). The cube pattern archive. http://www.cubeman.org


• Reid website (Reid M). http://www.cflmath.com/Rubik, for patterns see http://www.cflmath.com/Rubik/patterns.html


• Roux website (Roux G). http://www.grroux.free.fr


• Scherphius website (Scherphius J). Jaap Puzzles website http://www.jaapsch.net/puzzles/symmetr1.htm


• Speedsolving website. http://www.speedsolving.com/

• Storer website (Storer JA). http://www.cs.brandeis.edu/~storer/JimPuzzles/
  For Rubik cube, see: http://www.cs.brandeis.edu/~storer/JimPuzzles/
  RUBIK/Rubik3x3x3.pdf
  For puzzle book, see: http://www.cs.brandeis.edu/~storer/zzzJimPuzzles/
  JimPuzzlesBook.pdf

  www.math.ubc.ca/~cass/courses/m308/projects/rtran/rtran.pdf

  Cubism For Fun 15, p. 10 (Nederlandse Kubus Club) [cited from Wikipedia
  (Rubik’s cube)]

  be

• WCA (2016). World Cube Association Regulations. See §12 for notation.
  http://www.worldcubeadociation.org/regulations.htm

20 The code (rubikcube.sty)

The conventions we adopt regarding capital letters and the XYZ argument ordering
are detailed in Section 3.

Note that it is important when using a graphics package to use a trailing % on
the end of lines which break before the terminal curly bracket of a \newcommand.
This is to prevent accumulating spurious spaces which may otherwise appear in
figures and diagrams as a strange or unexpected horizontal shift or white-space.

20.1 Package heading

1 (*rubikcube)
2 \def\RCfileversion{5.0}\
3 \def\RCfiledate{2018/02/25}% February 25, 2018
4 \NeedsTeXFormat{LaTeX2e}
5 \ProvidesPackage{rubikcube}[]{RCfiledate\space (v\RCfileversion)}

The package requires TikZ—so we load it if not already loaded.
6 \@ifpackageloaded{tikz}{}{%
7 \typeout{---rubikcube requires the TikZ package.}%
8 \RequirePackage{tikz}%

The package requires the Forarray package (see Section 20.5)—so we load it if
not already loaded.
9 \@ifpackageloaded{forarray}{}{%
10 \typeout{---rubikcube requires the Forarray package.}%
11 \RequirePackage{forarray}%)
The package requires the IfThen package (see Section 20.5)—so we load it if not already loaded.
\@ifpackageloaded{ifthen}\%
\typeout{---rubikcube requires the IfThen package.}\%
\RequirePackage{ifthen}\%

First we create a suitable logo
\newcommand{rubikcube}{\textsc{rubikcube}}\%
\newcommand{Rubikcube}{\textsc{Rubikcube}}\%

20.2 Colours
We have adopted the following colour allocations—see Section 8 for details.
\definecolor{R}{HTML}{C41E33}\%
\definecolor{G}{HTML}{00BE38}\%
\definecolor{B}{HTML}{0051BA}\%
\definecolor{Y}{HTML}{FFFF00}\%
\colorlet{X}{black!30}% grey
\colorlet{O}{orange}\%
\colorlet{W}{white}\%

20.3 The rubikfont
We define two fonts for text associated with the Rubik glyphs (both the ‘arrow’ glyphs and the ‘letter’ glyphs), namely, (1) Computer Modern Sans (cmss), bold extended (bx), normal shape (n) at 10/12pt, and (2) a footnotesize (FNS) version (8pt) for the lower-case letters [for cmss see Latex Companion (2004), p. 417 & p. 354]. This has the effect of keeping the size of Rubik glyphs constant in the face of any changes in the document fonts. We make the baseline-skip values the same, since the ‘arrow’ glyphs generated by the \Rubik\ commands involve a single baseline-skip (for example, as with \Rubik{D}; see Section 20.21.13). We use the cmss font apostrophe as the ‘prime’ symbol (the user has the opportunity to use the maths \textprime instead—see Section 9.10).
\newcommand{rubikfont}{\fontsize{10}{12pt}\usefont{T1}{cmss}{bx}{n}}
\newcommand{rubikfontFNS}{\fontsize{8}{12pt}\usefont{T1}{cmss}{bx}{n}}
\newcommand{rubikprime}{'}\%

20.4 ShowCube command
The macro \ShowCube\{(minipage width)\}{\{TikZ scale factor\}\{\{Draw.. cmd\}\}} displays the cube inside a minipage, so that we can easily tailor the minipage width (#1) and also the TikZ scale factor (#2). The \ShowCubeF command places an fbox around the minipage so users can see the extent of any white space.
USAGE: \ShowCube\{2cm\}{0.5}\{\DrawRubikCubeRU\}

\footnote{Note the typo in Table 7.5 (p. 354): the font-series code for the Sans semi-bold condensed form is ‘sbd’ (not sbc).}
February 2017 (RWND): We first require a new length variable (which will become the minipage-width), so we can add the length 1.6pt to it (this is the width of the TikZ ultra-thick line which is used to draw the Rubik cubes). In order for a width of an image made up of \( x \) units to be equal to \( x \times (\text{scale-factor}) \) we need to add an extra line-width (i.e., to include the right-hand edge).

\[ \newlength\{\@showcubewidth\} \]

We can now build the two macros. We set the \texttt{\fboxsep} value to zero.

\[ \newcommand\{\ShowCube\}[3]\{ \]
\[ \setlength\{\fboxsep\}\{0\text{cm}\}% \]
\[ \setlength\{\@showcubewidth\}\{\#1\}% \]
\[ \advance\@showcubewidth\text{ by} 1.6\text{pt}\relax\% \]
\[ \begin{minipage}\{\the\@showcubewidth\} \]
\[ \centering \]
\[ \begin{tikzpicture}\[\text{scale=\#2}\] \]
\[ #3\% \]
\[ \end{tikzpicture}\% \]
\[ \end{minipage}\% \]
\[ \} \]
\[ \newcommand\{\ShowCubeF\}[3]\{ \]
\[ \setlength\{\fboxsep\}\{0\text{cm}\}% \]
\[ \setlength\{\fboxrule\}\{0.4\text{pt}\}% \]
\[ \setlength\{\@showcubewidth\}\{\#1\}% \]
\[ \advance\@showcubewidth\text{ by} 1.6\text{pt}\relax\% \]
\[ \framebox\{ \]
\[ \begin{minipage}\{\the\@showcubewidth\} \]
\[ \centering \]
\[ \begin{tikzpicture}\[\text{scale=\#2}\] \]
\[ #3\% \]
\[ \end{tikzpicture}\% \]
\[ \end{minipage}\% \]
\[ \} \]

\subsection{20.5 ShowSequence command}

The \texttt{\ShowSequence\{\{separator\}\}\{\{font-code\}\}\{\{sequence\}\}} command typesets a comma separated sequence of rotation commands. (See Section 11). This command takes three mandatory arguments: the first is the separator (\#1), the second is the font or style code (\#2), and third is a comma-separated sequence of Rubik rotation commands (\#3).

This command requires the \texttt{forarray} package—by Christian Schröppel—for the \texttt{\ForEachX} command and the \texttt{ifthen} package—by David Carlisle—for the \texttt{\ifthenelse} command. These two packages are loaded at startup if not already loaded. We first need to define two variables for use by the command; these are derived from the \texttt{forarray} package.

\[ \newcommand\{\x\}\{\thislevelitem\} \]
\[ \newcommand\{\xcount\}\{\thislevelcount\} \]
EXAMPLE USAGE: \ForEachX{;}{\texttt{x}}{L;R;U;D}

An important feature of the \ForEachX command is that it expands its third argument (the list of elements), since this allows the list of elements to be presented as a macro, which is extremely convenient for the user.

The \ShowSequence command typesets a sequence of elements (#3), and places an optional separator (#1) between them. For each element (x, see above) of the sequence #3 this command forms the construction #2{element of #3}. For example, if #2 = \rr, and D is an element of #3, then it will form the command \rr{D} etc.

Note that it is not straightforward to place the separator (#1) only between the derived elements (i.e., without the separator being either before the first element, or following the last element) using only the ForEachX command. This is because the ForEachX command processes each element in exactly the same way—i.e., a comma after the first element (good) means there will be a comma after the final element (bad).

We solve this problem by using the \ifthenelse command to allow the first element to be processed differently from all the remaining elements. This is because it is easy for \TeX{} to identify the first element of a sequence, but very difficult for it to identify the final element since we generally don’t know the number of elements beforehand. Consequently we identify the first element (using the \xcount variable (see above), and then process this first element without any comma; we then place a comma in front of each of the remaining elements.

We also create two fbox versions of the command: the ‘F’ version places an fbox about the whole output; the ‘f’ version places an fbox about each element in the output (these two versions can be helpful when checking white space).

Note: bugfix 22 October 2017 (RWDN): if the user implemented tt output using \tt instead of the standard \texttt as the #2 argument, then the action would not of course remain local, and consequently we have added a leading brace and complementary trailing brace around the the \ifthenelse... command in each of the following three macros to limit the action.

USAGE: \ShowSequence{,}{\rr}{R,L,Up,Dp.....}

\newcommand{\ShowSequence}{[3]{\ForEachX{,}{\ifthenelse{\xcount=1}{#2{x}}{#1#2{x}}}#3}}%
\newcommand{\ShowSequenceF}{[3]{\ForEachX{,}{\ifthenelse{\xcount=1}{#2{x}}{#1#2{x}}}fbox{#3}}}}%
\newcommand{\ShowSequencef}{[3]{\ForEachX{,}{\ifthenelse{\xcount=1}{fbox{#2{x}}}{#1fbox{#2{x}}}}#3}}%
SEQUENCE HOLDERS: providing none of the Rubik rotation-codes has a trailing integer (e.g., R3) then the Rubik macros \( \text{rr}, \text{rrh}, \text{Rubik}, \text{textRubik} \) will work as expected when used as the second argument in the \texttt{ShowSequence} command (described above). However, a problem arises when trying to process in this way any Rubik rotation-codes having a terminal integer (for example, short-codes e.g., R2, D3,...), since the \texttt{ShowSequence} macro cannot expand short-codes into their long-code elements (e.g., R,R,D,D,D,...).

Accommodating such codes when using the \texttt{ShowSequence} command is currently solved by using separate ‘holders’ for four derived strings, namely: \texttt{SequenceInfo}, \texttt{SequenceName}, \texttt{SequenceShort} and \texttt{SequenceLong} (for details see Section 11). These are generated automatically by the Perl RUBIKROTATION program, which returns a so-called ‘long’ version of the ‘short’ string (the argument of the \texttt{RubikRotation} command). For example, the Perl program converts any short codes (e.g., R2, D3,...) \( \rightarrow \) long form, e.g., R,R,D,D,D,... (see the RUBIKROTATION documentation for details). In order for the four ‘holders’ of these derived strings generated by the Perl program (written to the file \texttt{rubikstateNEW.dat}) to be accessible to the user they need to defined here so that they can then be ‘redefined’ (by the Perl program) in the file \texttt{rubikstateNEW.dat}:

70 \newcommand{\SequenceInfo}{\{} \% INFO only
71 \newcommand{\SequenceName}{\{} \% NAME only
72 \newcommand{\SequenceShort}{\{} \% original SHORT seq but with NO NAME
73 \newcommand{\SequenceLong}{\{} \% just the LONG string \& no name

20.6 SequenceBrace commands

The \texttt{SequenceBraceX\{\langle name\rangle\}\{\langle sequence\rangle\}} command is a tool for displaying a named sequence using a brace. The trailing A denotes that the brace is placed Above the sequence; B denotes the brace is Below the sequence. For usage see Section 12.

74 \newcommand{\SequenceBraceA}{\[2\]}{$\overbrace{\mbox{#2}}^{\mbox{#1}}$} \%
75 \newcommand{\SequenceBraceB}{\[2\]}{$\underbrace{\mbox{#2}}_{\mbox{#1}}$} \%
76 \newcommand{\SequenceBraceAF}{\[2\]}{$\fbox{\overbrace{\mbox{#2}}^{\mbox{#1}}}$} \%
77 \newcommand{\SequenceBraceBF}{\[2\]}{$\fbox{\underbrace{\mbox{#2}}_{\mbox{#1}}}$} \%

20.7 RubikFace commands

Cube face notation: U, D, L, R, F, B (Singmaster)
Cubie-facelet notation: t, m, b, l, m, r = top, middle, bottom, left, middle, right.
We use this lower case notation in XY-pairs to denote individual cubie facelets on a given face (to avoid confusion with cube Face notation), as follows:
- top row \( (1,2,3) = \text{lt, mt, rt} \)
- middle row \( (4,5,6) = \text{lm, mm, rm} \)
- bottom row \( (7,8,9) = \text{lb, mb, rb} \)
For example, the current colour of the right bottom facelet on the FRONT face is held as the variable \texttt{Fr b} etc.
The cubie-facelets (squares) on a face are also parameterized numerically \#1–\#9 reading from left-to-right, starting top-left & ending bottom-right, when used as arguments for specifying particular colours (as in the \RubikFace\ldots commands—see below).

These 5 commands allocate a colour to each of the 9 cubie-squares in the specified face (Up, Down, Left, Right, Front, Back). Each command takes 9 arguments (colour codes) in the order 1–9 as specified above.

**Example:** \RubikFaceUp{R}{O}{Y}{G}{B}{W}{X}{R}{G}

Each of the 9 \def{} commands below allocates one colour to a specific cubie-squares (facelet), using a simple three-letter encoding. Each letter is an initial letter of the words Up, Down, Left, Right, Front, Back, left, middle, right, top, middle, bottom.

For example, in the command \def\Urt{#1} the U denotes the Up face of the cube, while the rt denotes the “right-top” facelet on this face. Note that the order of the two lower-case letters (rt) is in the x,y order; i.e., the first of the two lower-case letters relates to the x direction (either left, middle, or right), while the second lower-case letter relates to the y direction (either top, middle, or bottom)—this rule makes it easy to remember the order.

\begin{verbatim}
\newcommand{\RubikFaceUp}{% \def\Ult{#1}\def\Umt{#2}\def\Urt{#3}\def\Ulm{#4}\def\Umm{#5}\def\Urm{#6}\def\Ulb{#7}\def\Umb{#8}\def\Urb{#9} % }
\newcommand{\RubikFaceDown}{% \def\Dlt{#1}\def\Dmt{#2}\def\Drt{#3}\def\Dlm{#4}\def\Dmm{#5}\def\Drm{#6}\def\Dlb{#7}\def\Dmb{#8}\def\Drb{#9} % }
\newcommand{\RubikFaceLeft}{% \def\Llt{#1}\def\Lmt{#2}\def\Lrt{#3}\def\Llm{#4}\def\Lmm{#5}\def\Lrm{#6}\def\Llb{#7}\def\Lmb{#8}\def\Lrb{#9} % }
\newcommand{\RubikFaceRight}{% \def\Rlt{#1}\def\Rmt{#2}\def\Rrt{#3}\def\Rlm{#4}\def\Rmm{#5}\def\Rrm{#6}\def\Rlb{#7}\def\Rmb{#8}\def\Rrb{#9} % }
\newcommand{\RubikFaceFront}{% \def\Flt{#1}\def\Fmt{#2}\def\Frt{#3}\def\Flm{#4}\def\Fmm{#5}\def\Frm{#6}\def\Flb{#7}\def\Fmb{#8}\def\Frb{#9} % }
\end{verbatim}
These 5 commands allocate the same colour to all 9 cubie-squares in the specified face (Up, Down, Left, Right, Front). Each command therefore takes only 1 argument (one of the colour codes).

For example, \RubikFaceUpAll{R}

Finally, we now use these commands to initialise all visible faces to the default colour grey (X)

\RubikFaceUpAll{X}
\RubikFaceDownAll{X}
\RubikFaceLeftAll{X}
\RubikFaceRightAll{X}
\RubikFaceFrontAll{X}
\RubikFaceBackAll{X}
20.8 RubikCubeGrey command

This command sets the face/colour configuration (state) of a 3x3x3 Rubik cube with no colours allocated except for the central cubie of each face. The colour configuration of the central cubies matches those defined for the RubikCubeSolved command (i.e., white opposite yellow etc). We also make WY and WB versions, and implement equivalent ‘gray’ versions (to be consistent with TikZ).

\newcommand{\RubikCubeGrey}{
\RubikFaceRight{X}{X}{X}{X}{G}{X}{X}{X}{X}\
\RubikFaceLeft {X}{X}{X}{X}{B}{X}{X}{X}{X}\
\RubikFaceUp {X}{X}{X}{X}{W}{X}{X}{X}{X}\
\RubikFaceDown {X}{X}{X}{X}{Y}{X}{X}{X}{X}\
\RubikFaceFront{X}{X}{X}{X}{O}{X}{X}{X}{X}\
\RubikFaceBack {X}{X}{X}{X}{R}{X}{X}{X}{X}\
}

\newcommand{\RubikCubeGray}{\RubikCubeGrey}
\newcommand{\RubikCubeGreyWY}{\RubikCubeGrey}
\newcommand{\RubikCubeGrayWY}{\RubikCubeGreyWY}

\%
\newcommand{\RubikCubeGreyWB}{
\RubikFaceRight{X}{X}{X}{X}{R}{X}{X}{X}{X}\
\RubikFaceLeft {X}{X}{X}{X}{O}{X}{X}{X}{X}\
\RubikFaceUp {X}{X}{X}{X}{W}{X}{X}{X}{X}\
\RubikFaceDown {X}{X}{X}{X}{B}{X}{X}{X}{X}\
\RubikFaceFront{X}{X}{X}{X}{G}{X}{X}{X}{X}\
\RubikFaceBack {X}{X}{X}{X}{Y}{X}{X}{X}{X}\
}
\newcommand{\RubikCubeGrayWB}{\RubikCubeGreyWB}

20.9 SolvedConfig command

\RubikSolvedConfig

This command sets the face/colour configuration (state) of a typical solved Rubik cube. Note that the order is Right, Left, Up, Down, Front, Back (i.e., X+, X−, Y+, Y−, Z+, Z−, order). We shall use this command to define solved cube configurations.

\newcommand{\RubikSolvedConfig}{[6]{%\
\RubikFaceRightAll{#1}\%\
\RubikFaceLeftAll{#2}\%\
\RubikFaceUpAll{#3}\%\
\RubikFaceDownAll{#4}\%\
\RubikFaceFrontAll{#5}\%\
\RubikFaceBackAll{#6}\%\
}}

20.10 RubikCubeGreyAll command

\RubikCubeGreyAll

This command sets the face/colour configuration (state) of a 3x3x3 Rubik cube with no colours allocated. This colour configuration is the same as the
startup default state—all cubies will appear grey. We implement it using the \RubikSolvedConfig command (above). We also implement an equivalent ‘grey’ version (to be consistent with TikZ).

\newcommand{\RubikCubeGreyAll}{\RubikSolvedConfig{X}{X}{X}{X}{X}{X}}%
\newcommand{\RubikCubeGrayAll}{\RubikCubeGreyAll}

20.11 RubikCubeSolved command

The first (default) command sets the face/colour configuration (state) one of the standard commercially available solved Rubik cube (white opposite yellow). The argument order follows the XYZ notation. For convenience we make a copy named \RubikCubeSolvedWY (denoting the White opposite Yellow configuration), and also a different version named \RubikCubeSolvedWB (denoting the White opposite Blue configuration). These represent the two standard versions of the Rubik Cube.

\newcommand{\RubikCubeSolved}{\RubikSolvedConfig{G}{B}{W}{Y}{O}{R}}%
\newcommand{\RubikCubeSolvedWY}{\RubikCubeSolved}%
\newcommand{\RubikCubeSolvedWB}{\RubikSolvedConfig{R}{O}{W}{B}{G}{Y}}%

20.12 Slice commands

These 6 commands allocate the colour arguments for the 6 visible ordered facelets along a horizontal slice. There are three horizontal slices to consider (Top, Middle, Bottom) and each has two viewpoints. The colour-code arguments are ordered 1–6 from left to right. The terminal L and R denote the Left (L) viewpoint and Right (R) viewpoint versions. Note that the two legacy ‘Equator’ versions (now replaced by ‘Middle’) are retained (below) to allow backward compatibility.

\newcommand{\RubikSliceTopR}[6]{%
\def\Flt{#1}\def\Fmt{#2}\def\Frt{#3}%
\def\Rlt{#4}\def\Rmt{#5}\def\Rrt{#6}%
}%
\newcommand{\RubikSliceTopL}[6]{%
\def\Llt{#1}\def\Lmt{#2}\def\Lrt{#3}%
\def\Flt{#4}\def\Fmt{#5}\def\Frt{#6}%
}%
\newcommand{\RubikSliceMiddleR}[6]{%
\def\Flm{#1}\def\Fmm{#2}\def\Frm{#3}%
\def\Rlm{#4}\def\Rmm{#5}\def\Rrm{#6}%
}%
\newcommand{\RubikSliceMiddleL}[6]{%
\def\Llm{#1}\def\Lmm{#2}\def\Lrm{#3}%
\def\Flm{#4}\def\Fmm{#5}\def\Frm{#6}%
}%
\newcommand{\RubikSliceEquatorR}[6]{%
\def\Flm{#1}\def\Fmm{#2}\def\Frm{#3}%
\def\Rlm{#4}\def\Rmm{#5}\def\Rrm{#6}%
}%
\newcommand{\RubikSliceEquatorL}[6]{%
20.13 Cube drawing macros

Since the three visible sides of a Rubik cube have up to 27 non-grey colours, and \TeX{} has only 9 macro parameters available, we are forced to draw Rubik cubes by first specifying the colours on each of the three faces, and then using a \texttt{DrawRubikCubeXY} command, where the trailing XY code defines the viewing direction (X = either R or L; Y = either U or D). The order of the XY code is important: X first, Y second (so it is easy to remember).

On each face the facetets are drawn in the following order: Top row (left to right), Middle row (left to right), Bottom row (left to right).

The TikZ draw cycle for each facetet square on a Rubik cube face cycles through the four corners of the facetet in the following order: lb \rightarrow lt \rightarrow rt \rightarrow rb; the code being lb (LeftBottom), lt (LeftTop), rt (RightTop), rb (RightBottom) (only need four coords); the (x,y) grid origin is at the bottom-left corner of the front face.

\begin{verbatim}
\newcommand{\DrawRubikCubeFrontFace}{%  
% ---top row left to right
\draw[line join=round,line cap=round,ultra thick,fill=\Flt]%  \( (0,2) \rightarrow (0,3) \rightarrow (1,3) \rightarrow (1,2) \rightarrow \) cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Fmt]%  \( (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (2,2) \rightarrow \) cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Frt]%  \( (2,2) \rightarrow (2,3) \rightarrow (3,3) \rightarrow (3,2) \rightarrow \) cycle;
% -----middle row left to right
\draw[line join=round,line cap=round,ultra thick,fill=\Flm]%  \( (0,1) \rightarrow (0,2) \rightarrow (1,2) \rightarrow (1,1) \rightarrow \) cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Fmm]%  \( (1,1) \rightarrow (1,2) \rightarrow (2,2) \rightarrow (2,1) \rightarrow \) cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Frm]%  \( (2,1) \rightarrow (2,2) \rightarrow (3,2) \rightarrow (3,1) \rightarrow \) cycle;
\end{verbatim}

This ‘\texttt{FrontFace}’ command is an ‘internal’ command which draws and paints all the facetets on the front face of a Rubik cube. It is used by all of the cube drawing macros which display the front face. The 9 colours are allocated by an earlier \texttt{RubikFaceFront} command. These Face macros are based, in part, on those of Peter Bartal (2011).
20.13.1 Viewing direction

The command ‘DrawRubikCubeXY’ command uses the trailing XY code to specify
the view direction (X = either R or L; Y = either U or D). The order of the XY
code is important: X first, Y second (so it is easy to remember).

\DrawRubikCubeRU

This command draws and paints a Rubik cube as viewed from the Right Upper
(RU) viewpoint. It starts by using the internal command \DrawRubikCubeFrontFace
to draw the FRONT face, and then draws the UP face followed by the RIGHT
face. The colours are allocated to particular facelets using the \RubikFaceUp,
\RubikFaceRight and \RubikFaceFront commands.

The (x,y) grid origin is at the bottom-left corner of the FRONT face (see Section
4). The perspective is designed so that the 2D graphic image of the side face
(RIGHT in this particular case) has its ‘horizontal’ lines running at 45 degrees.
This has the useful advantage that the 2D width of the side is exactly 1-unit, and
so makes it easy to determine the 2D (x,y) coordinates of any position, and hence
facilitates typesetting text onto the image of the cube using TikZ commands.
\begin{verbatim}
256 (2,3) -- (2.33,3.33) -- (3.33,3.33) -- (3,3) -- cycle;
257 \%\%------Right face------
258 \draw[line join=round,line cap=round,ultra thick,fill=\Rlt]%
259 (3,2) -- (3, 3) -- (3.33,3.33) -- (3.33,2.33) -- cycle;
260 \draw[line join=round,line cap=round,ultra thick,fill=\Rmt]%
261 (3.33,2.33) -- (3.33, 3.33) -- (3.66,3.66) -- (3.66,2.66) -- cycle;
262 \draw[line join=round,line cap=round,ultra thick,fill=\Rrt]%
263 (3.66,2.66) -- (3.66, 3.66) -- (4,4) -- (4,3) -- cycle;
264 \%\%--middle row
265 \draw[line join=round,line cap=round,ultra thick,fill=\Rlm]%
266 (3,1) -- (3, 2) -- (3.33,2.33) -- (3.33,1.33) -- cycle;
267 \draw[line join=round,line cap=round,ultra thick,fill=\Rmm]%
268 (3.33,1.33) -- (3.33, 2.33) -- (3.66,2.66) -- (3.66,1.66) -- cycle;
269 \draw[line join=round,line cap=round,ultra thick,fill=\Rrm]%
270 (3.66,1.66) -- (3.66, 2.66) -- (4,4) -- (4,2) -- cycle;
271 \%\%--bottom row
272 \draw[line join=round,line cap=round,ultra thick,fill=\Rlb]%
273 (3,0) -- (3, 1) -- (3.33,1.66) -- (3.33,0.66) -- cycle;
274 \draw[line join=round,line cap=round,ultra thick,fill=\Rmb]%
275 (3.33,0.66) -- (3.33, 1.66) -- (3.66,1.33) -- (3.66,0.33) -- cycle;
276 \draw[line join=round,line cap=round,ultra thick,fill=\Rrb]%
277 (3.66,0.66) -- (3.66, 1.66) -- (4,2) -- (4,1) -- cycle;
278 }
\end{verbatim}

\DrawRubikCube This command is equivalent to the previous \DrawRubikCubeRU and hence is the default form (i.e., if the trailing XY viewpoint code is accidentally omitted).

\newcommand{\DrawRubikCube}{\DrawRubikCubeRU}

\DrawRubikCubeRD This command draws and paints a Rubik cube as viewed from the Right Down (RD) viewpoint.

\newcommand{\DrawRubikCubeRD}{%
\DrawRubikCubeFrontFace \% frontface
\%\%------Right face------
\%\%--top row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlt]%
(3,2) -- (3, 3) -- (3.33,2.66) -- (3.33,1.66) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rmt]%
(3.33,1.66) -- (3.33, 2.66) -- (3.66,2.33) -- (3.66,1.33) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrt]%
(3.66,1.33) -- (3.66, 2.33) -- (4,2) -- (4,1) -- cycle;
\%\%--middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlm]%
(3,1) -- (3, 2) -- (3.33,1.33) -- (3.33,0.33) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rmm]%
(3.33,0.33) -- (3.33, 1.33) -- (3.66,1.66) -- (3.66,0.66) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrm]%
(3.66,0.66) -- (3.66, 1.66) -- (4,2) -- (4,1) -- cycle;
\%\%--bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlb]%
(3,0) -- (3, 1) -- (3.33,0.66) -- (3.33,0.33) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rmb]%
(3.33,0.33) -- (3.33, 0.66) -- (3.66,0.33) -- (3.66,0.06) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrb]%
(3.66,0.06) -- (3.66, 0.33) -- (4,2) -- (4,1) -- cycle;
This command draws and paints a Rubik cube as viewed from the Left Down (LD) viewpoint.
\DrawRubikCubeLU  This command draws and paints a Rubik cube as viewed from the Left Up (LU) viewpoint.

\newcommand{\DrawRubikCubeLU}{\%
\DrawRubikCubeFrontFace \%
\DrawRubikCubeBackFace \%
\DrawRubikCubeRightFace \%
\DrawRubikCubeLeftFace \%
\DrawRubikCubeUpFace \%
\DrawRubikCubeDownFace \%
}%
\begin{verbatim}
\draw[line join=round, line cap=round, ultra thick, fill=\Lrm] (-0.33,1.33) -- (-0.33, 2.33) -- (0,2) -- (0,1) -- cycle;
\%
\draw[line join=round, line cap=round, ultra thick, fill=\Llb] (-1,1) -- (-1, 2) -- (-0.66,1.66) -- (-0.66,0.66) -- cycle;
\draw[line join=round, line cap=round, ultra thick, fill=\Lmb] (-0.66,0.66) -- (-0.66, 1.66) -- (-0.33,1.33) -- (-0.33,0.33) -- cycle;
\draw[line join=round, line cap=round, ultra thick, fill=\Lrb] (-0.33,0.33) -- (-0.33, 1.33) -- (0,1) -- (0,0) -- cycle;
\%
\draw[line join=round, line cap=round, ultra thick, fill=\Ult] (-0.66,3.66) -- (-1, 4) -- (0,4) -- (0.33,3.66) -- cycle;
\draw[line join=round, line cap=round, ultra thick, fill=\Umt] (0.33,3.66) -- (0, 4) -- (1,4) -- (1.33,3.66) -- cycle;
\draw[line join=round, line cap=round, ultra thick, fill=\Urt] (1.33,3.66) -- (1, 4) -- (2,4) -- (2.33,3.66) -- cycle;
\%
\draw[line join=round, line cap=round, ultra thick, fill=\Ulm] (-0.33,3.33) -- (-0.66, 3.66) -- (0.33,3.66) -- (0.66,3.33) -- cycle;
\draw[line join=round, line cap=round, ultra thick, fill=\Umm] (0.66,3.33) -- (0.33, 3.66) -- (1.33,3.66) -- (1.66,3.33) -- cycle;
\draw[line join=round, line cap=round, ultra thick, fill=\Urm] (1.66,3.33) -- (1.33, 3.66) -- (2.33,3.66) -- (2.66,3.33) -- cycle;
\%
\draw[line join=round, line cap=round, ultra thick, fill=\Ulb] (0,3) -- (-0.33, 3.33) -- (0.66,3.33) -- (1,3) -- cycle;
\draw[line join=round, line cap=round, ultra thick, fill=\Umb] (1,3) -- (0.66, 3.33) -- (1.66,3.33) -- (2,3) -- cycle;
\draw[line join=round, line cap=round, ultra thick, fill=\Urb] (2,3) -- (1.66, 3.33) -- (2.66,3.33) -- (3,3) -- cycle;
\%
\end{verbatim}

RWDN19D removed DrawRubikLayerFace commands Feb 19 2018

### 20.14 DrawRubikFlatX commands

**BACKGROUND:** These commands (new in version 3.0) were modified from the earlier \FlatUp, \FlatDown etc., commands; i.e., they were renamed as a set of \Draw... commands so as to make this notation consistent with the other \Draw... commands. Note also that the \DrawRubikFace... commands are essentially these same commands but with their two coordinate arguments X,Y set to x = 0, y = 0—see Section 20.15

Each of these commands draws a separate (flat) face (9 facelets). Each command (except \DrawRubikFlatFront) takes two arguments, namely the X-coordinate and Y-coordinate of the bottom-left corner of the face. This (X,Y) pair of coordinates therefore allows the user to position the face in relation to the cube itself.
These commands were motivated by a need to be able to show hidden faces under certain circumstances.

Note also that the `\DrawRubikFlatFront` command takes no arguments, since by definition the bottom left corner of this face is at (0,0), and there seems to be no reason (just now) for this face to have the facility to be positioned otherwise.

**EXAMPLE:** The following command positions the Up face so that its bottom left corner is located at (0,3):

```
\DrawRubikFlatUp{0}{3}
```

These new commands are also used by the commands `\DrawRubikCubeF` and `\DrawRubikCubeSF` to draw various ‘flat’ representations of a Rubik cube.

The (x,y) variables are here encoded as (\ux, \uy) where the ‘u’ stands for Up etc. However, since we are unable to use a ‘dx, dy’ notation with the `\DrawRubikFlatDown` command (since dx and dy are already used by the `\cube@dx@dy@dz...` command), we encode these instead as (\ddx, \ddy).

```latex
\newcommand{\DrawRubikFlatUp}[2]{% 
  \pgfmathsetmacro{\ux}{#1}% 
  \pgfmathsetmacro{\uy}{#2}% 
  %--top row 
  \draw[line join=round,line cap=round,ultra thick,fill=\Ult] (\ux + 0,\uy + 2) -- (\ux + 0,\uy + 3) -- (\ux + 1,\uy + 3) -- (\ux + 1,\uy + 2) -- cycle; 
  \draw[line join=round,line cap=round,ultra thick,fill=\Umt] (\ux + 1,\uy + 2) -- (\ux + 1,\uy + 3) -- (\ux + 2,\uy + 3) -- (\ux + 2,\uy + 2) -- cycle; 
  \draw[line join=round,line cap=round,ultra thick,fill=\Urt] (\ux + 2,\uy + 2) -- (\ux + 2,\uy + 3) -- (\ux + 3,\uy + 3) -- (\ux + 3,\uy + 2) -- cycle; 
  %--middle row 
  \draw[line join=round,line cap=round,ultra thick,fill=\Ulm] (\ux + 0,\uy + 1) -- (\ux + 0,\uy + 2) -- (\ux + 1,\uy + 2) -- (\ux + 1,\uy + 1) -- cycle; 
  \draw[line join=round,line cap=round,ultra thick,fill=\Umm] (\ux + 1,\uy + 1) -- (\ux + 1,\uy + 2) -- (\ux + 2,\uy + 2) -- (\ux + 2,\uy + 1) -- cycle; 
  \draw[line join=round,line cap=round,ultra thick,fill=\Urm] (\ux + 2,\uy + 1) -- (\ux + 2,\uy + 2) -- (\ux + 3,\uy + 2) -- (\ux + 3,\uy + 1) -- cycle; 
  %--bottom row 
  \draw[line join=round,line cap=round,ultra thick,fill=\Ulb] (\ux + 0,\uy + 0) -- (\ux + 0,\uy + 1) -- (\ux + 1,\uy + 1) -- (\ux + 1,\uy + 0) -- cycle; 
  \draw[line join=round,line cap=round,ultra thick,fill=\Umb] (\ux + 1,\uy + 0) -- (\ux + 1,\uy + 1) -- (\ux + 2,\uy + 1) -- (\ux + 2,\uy + 0) -- cycle; 
  \draw[line join=round,line cap=round,ultra thick,fill=\Urb] (\ux + 2,\uy + 0) -- (\ux + 2,\uy + 1) -- (\ux + 3,\uy + 1) -- (\ux + 3,\uy + 0) -- cycle;
```

\newcommand\DrawRubikFlatDown[2][]{\pgfmathsetmacro\ddx{#1}\pgfmathsetmacro\ddy{#2}\begin{scope}%
%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Dlt] (\ddx + 0,\ddy + 2) -- (\ddx + 0,\ddy + 3) -- (\ddx + 1,\ddy + 3) -- (\ddx + 1,\ddy + 2) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Dmt] (\ddx + 1,\ddy + 2) -- (\ddx + 1,\ddy + 3) -- (\ddx + 2,\ddy + 3) -- (\ddx + 2,\ddy + 2) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Drt] (\ddx + 2,\ddy + 2) -- (\ddx + 2,\ddy + 3) -- (\ddx + 3,\ddy + 3) -- (\ddx + 3,\ddy + 2) -- cycle;
%-----middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Dlm] (\ddx + 0,\ddy + 1) -- (\ddx + 0,\ddy + 2) -- (\ddx + 1,\ddy + 2) -- (\ddx + 1,\ddy + 1) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Dmm] (\ddx + 1,\ddy + 1) -- (\ddx + 1,\ddy + 2) -- (\ddx + 2,\ddy + 2) -- (\ddx + 2,\ddy + 1) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Drm] (\ddx + 2,\ddy + 1) -- (\ddx + 2,\ddy + 2) -- (\ddx + 3,\ddy + 2) -- (\ddx + 3,\ddy + 1) -- cycle;
%----bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Dlb] (\ddx + 0,\ddy + 0) -- (\ddx + 0,\ddy + 1) -- (\ddx + 1,\ddy + 1) -- (\ddx + 1,\ddy + 0) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Dmb] (\ddx + 1,\ddy + 0) -- (\ddx + 1,\ddy + 1) -- (\ddx + 2,\ddy + 1) -- (\ddx + 2,\ddy + 0) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Drb] (\ddx + 2,\ddy + 0) -- (\ddx + 2,\ddy + 1) -- (\ddx + 3,\ddy + 1) -- (\ddx + 3,\ddy + 0) -- cycle;
\end{scope}}
\newcommand\DrawRubikFlatLeft[2][]{\pgfmathsetmacro\lx{#1}\pgfmathsetmacro\ly{#2}\begin{scope}%
%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Llt] (\lx + 0,\ly + 2) -- (\lx + 0,\ly + 3) -- (\lx + 1,\ly + 3) -- (\lx + 1,\ly + 2) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lmt] (\lx + 1,\ly + 2) -- (\lx + 1,\ly + 3) -- (\lx + 2,\ly + 3) -- (\lx + 2,\ly + 2) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lrt] (\lx + 0,\ly + 2) -- (\lx + 2,\ly + 2) -- (\lx + 3,\ly + 2) -- (\lx + 3,\ly + 1) -- cycle;
%-----middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Llm] (\lx + 0,\ly + 1) -- (\lx + 0,\ly + 2) -- (\lx + 1,\ly + 2) -- (\lx + 1,\ly + 1) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lmm] (\lx + 1,\ly + 1) -- (\lx + 1,\ly + 2) -- (\lx + 2,\ly + 2) -- (\lx + 2,\ly + 1) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lrm] (\lx + 2,\ly + 1) -- (\lx + 2,\ly + 2) -- (\lx + 3,\ly + 2) -- (\lx + 3,\ly + 1) -- cycle;
%----bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Llb] (\lx + 0,\ly + 0) -- (\lx + 0,\ly + 1) -- (\lx + 1,\ly + 1) -- (\lx + 1,\ly + 0) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lmb] (\lx + 1,\ly + 0) -- (\lx + 1,\ly + 1) -- (\lx + 2,\ly + 1) -- (\lx + 2,\ly + 0) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lrb] (\lx + 2,\ly + 0) -- (\lx + 2,\ly + 1) -- (\lx + 3,\ly + 1) -- (\lx + 3,\ly + 0) -- cycle;
\end{scope}}
\DrawRubikCubeF Draws a standard flat (F) representation of the Rubik cube (colours only). Note that \DrawRubikFlatFront (below) does not take any arguments (x,y).

\DrawRubikCubeSF Draws a Rubik cube together with the three hidden faces (colours only) in a semi-flat (SF) representation. The (x,y) arguments (below) are for the bottom-left corner of the face.

20.14.1 DrawRubikFaceXSide commands

These six commands draw a face together with all four sidebars (colours only). They use the \DrawRubikFlatX commands to draw the face, and the \side@bar.
commands to draw the sidebars. Since each of the four sides of a face has three small (side) bars, we use a total of 12 side@bar commands for each command.

The parameter codes for the \side@bar.. commands are as follows (see Section 20.16.1 for full details). Top (T) and Bottom (B) = left to right; Left (L) and Right (R) = top to bottom:

Note that since these commands are quite long, they all have a slightly more convenient short-hand version whereby the terminal two words are contracted to their two initial letters (see Section 20.15).

\DrawRubikFaceUpSide \DrawRubikFaceFrontSide

\newcommand{\DrawRubikFaceUpSide}{\%
\DrawRubikFlatUp{0}{0}%
\side@barT{1}{\Brt}
\side@barT{2}{\Bmt}
\side@barT{3}{\Blt}
\side@barL{1}{\Lrt}
\side@barL{2}{\Lmt}
\side@barL{3}{\Llt}
\side@barR{1}{\Rlt}
\side@barR{2}{\Rmt}
\side@barR{3}{\Rrt}
\side@barB{1}{\Flt}
\side@barB{2}{\Fmt}
\side@barB{3}{\Frt}
}\}

\newcommand{\DrawRubikFaceFrontSide}{\%
\DrawRubikFlatFront{0}{0}%
\side@barT{1}{\Ulb}
\side@barT{2}{\Umb}
\side@barT{3}{\Urb}
\side@barL{1}{\Lrb}
\side@barL{2}{\Lrm}
\side@barL{3}{\Lrt}
\side@barR{1}{\Rlb}
\side@barR{2}{\Rlm}
\side@barR{3}{\Rlt}
\side@barB{1}{\Dlt}
\side@barB{2}{\Dmt}
\side@barB{3}{\Drt}
\DrawRubikFaceRightSide \DrawRubikFaceLeftSide \DrawRubikFaceBackSide

\DrawRubikFaceRightSide
\newcommand{\DrawRubikFaceRightSide}{\DrawRubikFlatRight{0}{0}\%\side@barT{1}{\Urb}\side@barT{2}{\Urm}\side@barT{3}{\Urt}\%\side@barL{1}{\Frb}\side@barL{2}{\Frm}\side@barL{3}{\Frt}\%\side@barR{1}{\Blb}\side@barR{2}{\Blm}\side@barR{3}{\Blt}\%\side@barB{1}{\Drt}\side@barB{2}{\Drm}\side@barB{3}{\Drb}\%}

\DrawRubikFaceLeftSide
\newcommand{\DrawRubikFaceLeftSide}{\DrawRubikFlatLeft{0}{0}\%\side@barT{1}{\Ult}\side@barT{2}{\Ulm}\side@barT{3}{\Ulb}\%\side@barL{1}{\Brb}\side@barL{2}{\Brm}\side@barL{3}{\Brt}\%\side@barR{1}{\Flb}\side@barR{2}{\Flm}\side@barR{3}{\Flt}\%\side@barB{1}{\Dlb}\side@barB{2}{\Dlm}\side@barB{3}{\Dlt}\%}

\DrawRubikFaceBackSide
\newcommand{\DrawRubikFaceBackSide}{\DrawRubikFlatBack{0}{0}\%\side@barT{1}{\Urt}\%

Draws the RIGHT face together with all four sidebars.

Draws the LEFT face together with all four sidebars.

Draws the BACK face together with all four sidebars.
\DrawRubikFaceDownSide \DrawRubikFace commands

\DrawRubikFlatDown{0}{0} \DrawRubikFlat commands were originally just \DrawFlat commands. They were subsequently copied & renamed (here) as the more intuitive \DrawFace commands, since most of the time the user wants just to draw a particular face with or without sidebars.

In v5 it was necessary to rename them as \DrawRubikFace commands since they relate to a 3x3x3 cube. At the same time we also included their more convenient short-hand versions (eg \DrawRubikFaceUpSide \DrawRubikFaceUS etc), as the commands were getting a bit too long.

We therefore now deprecate the use of the earlier \DrawFace commands,
although they will be maintained for the moment at least.

\newcommand{\DrawRubikFaceUp}{\DrawRubikFlatUp{0}{0}}
\newcommand{\DrawRubikFaceDown}{\DrawRubikFlatDown{0}{0}}
\newcommand{\DrawRubikFaceLeft}{\DrawRubikFlatLeft{0}{0}}
\newcommand{\DrawRubikFaceRight}{\DrawRubikFlatRight{0}{0}}
\newcommand{\DrawRubikFaceFront}{\DrawRubikFlatFront{0}{0}}
\newcommand{\DrawRubikFaceBack}{\DrawRubikFlatBack{0}{0}}
\newcommand{\DrawRubikFaceU}{\DrawRubikFaceUp}
\newcommand{\DrawRubikFaceD}{\DrawRubikFaceDown}
\newcommand{\DrawRubikFaceL}{\DrawRubikFaceLeft}
\newcommand{\DrawRubikFaceR}{\DrawRubikFaceRight}
\newcommand{\DrawRubikFaceF}{\DrawRubikFaceFront}
\newcommand{\DrawRubikFaceB}{\DrawRubikFaceBack}
\newcommand{\DrawRubikFaceUS}{\DrawRubikFaceUpSide}
\newcommand{\DrawRubikFaceDS}{\DrawRubikFaceDownSide}
\newcommand{\DrawRubikFaceLS}{\DrawRubikFaceLeftSide}
\newcommand{\DrawRubikFaceRS}{\DrawRubikFaceRightSide}
\newcommand{\DrawRubikFaceFS}{\DrawRubikFaceFrontSide}
\newcommand{\DrawRubikFaceBS}{\DrawRubikFaceBackSide}

RWDN19A Finally, we create the short-hand versions.

\newcommand{\DrawRubikFaceU}{\DrawRubikFaceUp}
\newcommand{\DrawRubikFaceD}{\DrawRubikFaceDown}
\newcommand{\DrawRubikFaceL}{\DrawRubikFaceLeft}
\newcommand{\DrawRubikFaceR}{\DrawRubikFaceRight}
\newcommand{\DrawRubikFaceF}{\DrawRubikFaceFront}
\newcommand{\DrawRubikFaceB}{\DrawRubikFaceBack}
\newcommand{\DrawRubikFaceUS}{\DrawRubikFaceUpSide}
\newcommand{\DrawRubikFaceDS}{\DrawRubikFaceDownSide}
\newcommand{\DrawRubikFaceLS}{\DrawRubikFaceLeftSide}
\newcommand{\DrawRubikFaceRS}{\DrawRubikFaceRightSide}
\newcommand{\DrawRubikFaceFS}{\DrawRubikFaceFrontSide}
\newcommand{\DrawRubikFaceBS}{\DrawRubikFaceBackSide}

20.16 Sidebars (Face)

Sidebar commands draw narrow bars of colour indicating the side colours of each
of the facelets forming the side of a given layer (face). Each Sidebar is the length
of a single facelet (see Section 10.6).

These three commands allow the user to set the Width, Length and Separation
parameters for the sidebar (in decimal values, where 1 is equivalent to the length
of the side of a facelet).

\newcommand{\RubikSidebarWidth}[1]{\pgfmathsetmacro{\bw}{#1}}
\newcommand{\RubikSidebarLength}[1]{\pgfmathsetmacro{\bl}{#1}}
\newcommand{\RubikSidebarSep}[1]{\pgfmathsetmacro{\bs}{#1}}

We first set some default values

\RubikSidebarWidth{0.3}\%
\RubikSidebarLength{1}\%
\RubikSidebarSep{0.3}\%

In order to avoid conflicting nomenclature (between bar and Bar) we recommend
using the lowercase ‘bar’ and deprecate the use of ‘Bar’ in commands. For back-
wards compatibility, however, we will retain the three original ‘Bar’ commands
(for the moment at least), as follows:

\newcommand{\RubikSidebarWidth}[1]{\pgfmathsetmacro{\bw}{#1}}
\newcommand{\RubikSidebarLength}[1]{\pgfmathsetmacro{\bl}{#1}}
\newcommand{\RubikSidebarSep}[1]{\pgfmathsetmacro{\bs}{#1}}
20.16.1 Drawing and allocating a colour to a single facelet sidebar

Full length face sidebars are really multiple instances of small single facelet bars, each of which is drawn using one of the internal sidebar commands.

There are four \texttt{\textbackslash side@barX} commands, each with a trailing letter code X, which indicates the position of the sidebar relative to the square face displayed, namely either T (Top), B (Bottom), L (Left) or R (Right)—see Figure 5. Each \texttt{\textbackslash side@barX} command takes two arguments: The first argument is a number (distance) \{1 | 2 | 3\} from the relevant axis depending on whether it is a vertical or horizontal sidebar (e.g., 1 = first bar (nearest the origin); 2 = second bar, 3 = third bar).

The second argument is the facelet location-code expressed as a command (e.g., \texttt{\textbackslash Lrt}, \texttt{\textbackslash Dlb} etc. Note that the facelet location uses a three letter code: the first (capital) letter (U, D, L, R, F, B) denotes the face; the second (lower-case) letter (l,m,r) is the ‘x’ position in the 3x3 matrix; the third (lower-case) letter (t,m,b) is the ‘y’ position—see Figure 5.

Example: the following command draws a small single Right sidebar, in the middle position (no. 2), with the colour allocated to the Rlt facelet (left top facelet in the right face).

\texttt{\textbackslash side@barR\{2\}\{\textbackslash Rlt\}}

Notice that we use the command \texttt{\textbackslash Rlt} as the argument; this is because the command is defined as: \texttt{\textbackslash def\{\textbackslash Rlt\}\{#1\}}, and hence the command gets replaced by the colour-code currently allocated to this particular facelet.

There are three small rectangular sidebars on each of the four sides of a 3x3 square face, and these are embedded in a coordinate system with origin at the bottom left corner (0,0) of the square face (see Figure 5).

The \texttt{\textbackslash side@bar..} command also implements the set (or default) RubikSidebarLength \texttt{\textbackslash bl}, RubikSidebarWidth \texttt{\textbackslash bw} and RubikSidebarSep \texttt{\textbackslash bs} (separation) values mentioned above. \texttt{\textbackslash blh = \textbackslash Half \textbackslash bl = \textbackslash bl/2}. Note that the TikZ \texttt{\textbackslash pgfmathsetmacro} commands (which do the maths) must be inside the \texttt{\textbackslash side@bar..} command in order to work. The start point of the TikZ \texttt{\textbackslash draw} command for each rectangular sidebar is the bottom Left corner of the sidebar = (\texttt{\textbackslash dx,\textbackslash dy}).
The \NoSidebar\{\langle colour code\rangle\} command defines a (single) colour for which sidebars should not be drawn (particularly useful when drawing OLL configurations). This idea was suggested by Robert Marík (May 2017) — see Section 10.8. The principle is that we let the command \NoSidebar define a face colour, and then we use the \ifthenelse{\equal{\#2}{\NoSidebar}}{\ldots} structure inside the \sidebar\ldots commands (see below) to either (a) draw all sidebars as usual (if \NoSidebar is undefined), or (b) draw all sidebars except those having the \NoSidebar colour (if \NoSidebar colour = \#2).

Usage: \NoSidebar\{X\} If this command in not inside an environment, then its action will continue until it is disabled (undefined) as follows: \NoSidebar{\}.

\begin{verbatim}
\def\no@sidebar{}%
\newcommand{\NoSidebar}[1]{\def\no@sidebar{#1}}
\newcommand{\sidebarL}[2]{% #1 = cubie possn no, #2 = colour \pgfmathsetmacro{\blh}{\bl*(0.5)}\% \pgfmathsetmacro{\dx}{0 - \bs - \bw}\% \pgfmathsetmacro{\dy}{#1-1+0.5-\blh}\% \ifthenelse{\equal{\#2}{\no@sidebar}}{}{\pgfmathsetmacro{\dx}{0 - \bs - \bw}\% \pgfmathsetmacro{\dy}{#1-1+0.5-\blh}\% \draw[fill=#2] (\dx,\dy) -- (\dx,\dy+\bl) -- (\dx+\bw,\dy+\bl) -- (\dx+\bw,\dy) -- cycle;}
\newcommand{\sidebarR}[2]{% #1 = cubie possn no, #2 = colour \pgfmathsetmacro{\blh}{\bl*(0.5)}\% \pgfmathsetmacro{\dx}{3 + \bs}\% \pgfmathsetmacro{\dy}{#1 -1+0.5-\blh}\% \ifthenelse{\equal{\#2}{\no@sidebar}}{}{\pgfmathsetmacro{\dx}{3 + \bs}\% \pgfmathsetmacro{\dy}{#1 -1+0.5-\blh}\% \draw[fill=#2] (\dx,\dy) -- (\dx,\dy+\bl) -- (\dx+\bw,\dy+\bl) -- (\dx+\bw,\dy) -- cycle;}
\newcommand{\sidebarT}[2]{% #1 = cubie possn no, #2 = colour \pgfmathsetmacro{\blh}{\bl*(0.5)}\% \pgfmathsetmacro{\dx}{#1 -1+0.5-\blh}\% \pgfmathsetmacro{\dy}{3 + \bs}\% \ifthenelse{\equal{\#2}{\no@sidebar}}{}{\pgfmathsetmacro{\dx}{#1 -1+0.5-\blh}\% \pgfmathsetmacro{\dy}{3 + \bs}\% \draw[fill=#2] (\dx,\dy) -- (\dx,\dy+\bw) -- (\dx+\bl,\dy+\bw) -- (\dx+\bl,\dy) -- cycle;}
\newcommand{\sidebarB}[2]{% #1 = cubie possn no, #2 = colour \pgfmathsetmacro{\blh}{\bl*(0.5)}\% \pgfmathsetmacro{\dx}{#1 -1+0.5-\blh}\% \pgfmathsetmacro{\dy}{0 - \bs- \bw}\% \ifthenelse{\equal{\#2}{\no@sidebar}}{}{\pgfmathsetmacro{\dx}{0 - \bs- \bw}\% \pgfmathsetmacro{\dy}{0 - \bs- \bw}\% \draw[fill=#2] (\dx,\dy) -- (\dx,\dy+\bw) -- (\dx+\bl,\dy+\bw) -- (\dx+\bl,\dy) -- cycle;}
\end{verbatim}
20.17 Sidebars (Cube)

In order to position sidebars adjacent to a Rubik Cube (i.e., in 3D) requires that we first make some new \texttt{\textbackslash side@bar...} commands for drawing sidebars adjacent to the back face of the cube (we have already made the macros for the front face sidebars—see Section 20.16). Furthermore, these new macros need to be tailored to each of the four standard cube viewing directions RU, LU, RD, LD.

Finally, the USER commands for drawing these sidebars need to accommodate (a) some code for identifying each set of sidebars, and (b) the viewing direction. So, for example, a USER command for drawing the sidebars associated with the cube edge formed by the right face and the back face (let's define this as the RB sidebar) as viewed from the RU direction, might be something like \texttt{\textbackslash DrawRubikCubeSidebarRB{RU}}. Since this is not particularly user-friendly, we can improve on this slightly for the USER by (a) defining the sidebar as \texttt{SidebarRB}, and (b) appending the view direction in a curly bracket, say as \{RU\}. This allows a more intuitive command structure for the USER, as follows: \texttt{\textbackslash DrawRubikCubeSidebarRB{RU}}. We then use the \texttt{\textbackslash @join} command to append the string RU to the string \texttt{DrawRubikCubeSidebarRB} forming the (internal) command \texttt{\textbackslash DrawRubikCubeSidebarRB{RU}}.

In the following we will group the code according to the view direction (RU, LU, RD, LD).

20.17.1 Sidebars: RU view

Right-Back vert sidebar (RU view)

Need to write a new command for this position modified from \texttt{\textbackslash side@barR} (in Section 20.16.1) draws only a single small bar each of the three small bars has a numbered position (1,2,3); \((dx,dy)\) = bottom Left corner of single facelet bar

\begin{verbatim}
\newcommand{\side@barRubikRbackRU}[2]{{% 
  \% #1 = cubie possn no, #2 = colour 
  \% dx --> dx+1 
  \% dy --> dy+1 
  \ifthenelse{\equal{#2}{\no@sidebar}}{}{% 
    \pgfmathsetmacro{\blh}{\bl*(0.5)}% 
    \pgfmathsetmacro{\dx}{3 + \bs +1}% 
    \pgfmathsetmacro{\dy}{#1 -1+0.5-\blh +1}% 
    \draw[fill=#2] (\dx,\dy) -- (\dx,\dy + \bl) -- (\dx+\bw,\dy+\bl) -- (\dx+\bw,\dy) -- cycle; 
  }}
\end{verbatim}

Make the RB (RightBack) version; bar 1 is at the bottom

\begin{verbatim}
\newcommand{\DrawRubikCubeSidebarRB{RU}}{% 
\side@barRubikRbackRU{3}{\Blt}% 
\side@barRubikRbackRU{2}{\Blm}% 
\side@barRubikRbackRU{1}{\Blb}% 
\end{verbatim}
Now do the reverse (BR) = RB

\newcommand{\DrawRubikCubeSidebarBRU}{\DrawRubikCubeSidebarRBRU}

Make the join commands
\newcommand{\DrawRubikCubeSidebarBR}[1]{\@join{\DrawRubikCubeSidebarBR}{#1}}
\newcommand{\DrawRubikCubeSidebarRB}[1]{\@join{\DrawRubikCubeSidebarRB}{#1}}

**Up-Back horiz sidebar (RU view)**

Need to write a new command for this position modified from \side@barT (in Section 20.16.1) draws only a single small bar each of the three small bars has a numbered position (1,2,3); (dx,dy) = bottom Left corner of single facelet bar

\newcommand{\side@barRubikTbackRU}[2]{% 
  \ifthenelse{\equal{#2}{\no@sidebar}}{}{% 
    \pgfmathsetmacro{\blh}{\bl*(0.5)}% 
    \pgfmathsetmacro{\dx}{#1 -1+0.5-\blh +1}% 
    \pgfmathsetmacro{\dy}{3 +\bs +1}% 
    \draw[fill=#2](\dx,\dy) -- (\dx,\dy + \bw) -- (\dx+\bl,\dy + \bw) -- (\dx+\bl,\dy) -- cycle; 
  }}%}

Make the UB (Up-Back) version; bar 1 is at the left, 3 on the rhs (as we look at the image)
\newcommand{\DrawRubikCubeSidebarUBRU}{% 
  \side@barRubikTbackRU{1}{\Brt}\side@barRubikTbackRU{2}{\Bmt}\side@barRubikTbackRU{3}{\Blt}}%}

Now do the reverse (BU) = UB
\newcommand{\DrawRubikCubeSidebarUBRU}{\DrawRubikCubeSidebarUBRU}

Make the join commands
\newcommand{\DrawRubikCubeSidebarUB}[1]{\@join{\DrawRubikCubeSidebarUB}{#1}}
\newcommand{\DrawRubikCubeSidebarBU}[1]{\@join{\DrawRubikCubeSidebarBU}{#1}}

**Front-Left vert sidebar (RU view)**

For the front face we can use the regular \side@barL commands since it is the same as for an ordinary face sidebar
\newcommand{\DrawRubikCubeSidebarFLRU}{% 
  \side@barL{3}{\Lrt}\side@barL{2}{\Lrm}\side@barL{1}{\Lrb}}%}
Now do the reverse (LF)
\newcommand{\DrawRubikCubeSidebarLFRU}{\DrawRubikCubeSidebarFLRU}

Now do the two join commands
\newcommand{\DrawRubikCubeSidebarFL}{\@join{\DrawRubikCubeSidebarFL}{#1}}\newcommand{\DrawRubikCubeSidebarLF}{\@join{\DrawRubikCubeSidebarLF}{#1}}

Front-Down horizontal sidebar (RU view)
Horiz sidebar, so 1 at the left, 2=middle, 3=rhs) here we have to use the B for
bottom (of front face) and the facetets of the top row of the Down face
\newcommand{\DrawRubikCubeSidebarFDRU}{\side@barB{1}{\Dlt}}\side@barB{2}{\Dmt}\side@barB{3}{\Drt}

Now do the reverse (DF) = FD
\newcommand{\DrawRubikCubeSidebarDFRU}{\DrawRubikCubeSidebarFDRU}

Now do the two join commands
\newcommand{\DrawRubikCubeSidebarFD}{\@join{\DrawRubikCubeSidebarFD}{#1}}\newcommand{\DrawRubikCubeSidebarDF}{\@join{\DrawRubikCubeSidebarDF}{#1}}
But FD-LU is the same as FD-RU, so need to make copies of each
\newcommand{\DrawRubikCubeSidebarDFLU}{\DrawRubikCubeSidebarDFRU}\newcommand{\DrawRubikCubeSidebarFDLU}{\DrawRubikCubeSidebarFDRU}

20.17.2 Sidebars: LU view
Left-Back vert sidebar (LU view)
Need to write a new command for this position modified from \side@barL (in
Section 20.16.1) draws only a single small bar each of the three small bars has a
numbered position (1,2,3): (dx,dy) = bottom Left corner of single facelet bar
\newcommand{\side@barRubikLbackLU}{\% #1 = cubie possn no, #2 = colour
\% dx --> dx-1 \% dy --> dy+1 \ifthenelse{\equal{#2}{\no@sidebar}}{}{\pgfmathsetmacro{\bl}{(0.5)}\pgfmathsetmacro{\dx}{0 - \bs - \bw -1}\pgfmathsetmacro{\dy}{#1 -1+0.5-\bl+1}\draw[fill=#2] (\dx,\dy) -- (\dx,\dy + \bl) -- (\dx+\bw,\dy + \bl) -- (\dx+\bw,\dy) -- cycle;}}
Make the LB (LeftBack) version; bar 1 is at the bottom
\newcommand{\DrawRubikCubeSidebarBLvL}\{% 
\side@barRubikLbackLU\{1\}\{\Brt\}% 
\side@barRubikLbackLU\{2\}\{\Bmt\}% 
\side@barRubikLbackLU\{3\}\{\Blt\}% 
%}

Now do the reverse (BL) = LB
\newcommand{\DrawRubikCubeSidebarBLvL}{\DrawRubikCubeSidebarBLvL}

Make the join commands
\newcommand{\DrawRubikCubeSidebarLBvL}{\@join{\DrawRubikCubeSidebarLBvL}{#1}}
\newcommand{\DrawRubikCubeSidebarBLvL}{\@join{\DrawRubikCubeSidebarBLvL}{#1}}

Up-Back horizontal sidebar (LU view)
Modified from \side@barT (in Section 20.16.1) draws only a single small bar each of the three small bars has a numbered position (1,2,3); (dx,dy) = bottom Left corner of single facelet bar
\newcommand{\side@barRubikTbackLU}{% 
%%% #1 = cubie possn no; #2 = colour
%%% dx --> dx-1
%%% dy --> dy+1
% \ifthenelse{equal\{#2\}\{\no@sidebar\}}{}{%
% \pgfmathsetmacro\{blh\}\{bl*\{0.5\}\%
% \pgfmathsetmacro\{dx\}\{#1 -1+0.5-\{blh \}-1\%
% \pgfmathsetmacro\{dy\}\{3 +\bs +1\%
% \draw[fill=#2] \{(dx,dy) \--) \{(dx,dy + \bw)\} -- \{(dx+\bl,dy) \--) \{(dx+\bl,dy) \--) cycle;\%
% }}

Make the UB (Up-Back) version; bar 1 is at the left, 3 on the rhs (as we look at the image)
\newcommand{\DrawRubikCubeSidebarUBLvL}{% 
\side@barRubikTbackLU\{1\}\{\Brt\}% 
\side@barRubikTbackLU\{2\}\{\Bmt\}% 
\side@barRubikTbackLU\{3\}\{\Blt\}% 
%}

Now do the reverse (BU) = UB
\newcommand{\DrawRubikCubeSidebarUBLvL}{\DrawRubikCubeSidebarUBLvL}

Do not need to make the join commands as the USER commands for BU and UB are the same as for the RU.

Front-Right vertical sidebar (LU view)
Only needed for the LU view and LD. for the front face we can use the regular side@barR commands since it is the same as for an ordinary face sidebar RHS
20.17.3 Sidebars: RD view

Front-Up horizontal sidebar (RD view)
Horiz sidebar, so 1 at the left, 2=middle, 3= rhs here we have to use the T for bottom (of front face) and the facelets of the top row of the Down face
Now do the reverse (UF) = FU
Now do the two join commands

Front-Left vertical sidebar (RD view)
Front LEFT (RD view = same as for RU)

Right-Back vertical sidebar (RD view)
Modified from \side@barR (in Section 20.16.1) draws only a single small bar each of the three small bars has a numbered position (1,2,3); (dx,dy) = bottom Left corner of single facelet bar
Make the RB (RightBack) version; bar 1 is at the bottom
\newcommand{\DrawRubikCubeSidebarRBRD}{\side@barRubikRbackRD{3}{\Blt}\side@barRubikRbackRD{2}{\Blm}\side@barRubikRbackRD{1}{\Blb}}

Now do the reverse (BR) = RB
\newcommand{\DrawRubikCubeSidebarBRRD}{\DrawRubikCubeSidebarRBRD}

Do NOT need to make the join commands (same as for the RU view)

**Down-Back horizontal sidebar (RD view)**
Modified from \side@barB (in Section 20.16.1) draws only a single small bar each of the three small bars has a numbered position (1,2,3); (dx,dy) = bottom Left corner of single facelet bar
\newcommand{\side@barRubikBbackRD}[2]{%% #1 = cubie possn no; #2 = colour
%% dx --> dx+1
%% dy --> dy-1
\ifthenelse{\equal{#2}{\no@sidebar}}{}{\pgfmathsetmacro{\blh}{\bl*(0.5)}\pgfmathsetmacro{\dx}{#1 -1+0.5-\blh+1}\pgfmathsetmacro{\dy}{0-\bs-\bw-1}\draw[fill=#2] (\dx,\dy) -- (\dx,\dy+\bw) -- (\dx+\bw,\dy) -- (\dx+\bw,\dy+\bw) -- cycle;}}

Make the DB (Down-Back) version; bar 1 is at the left, 3 on the rhs (as we look at the image)
\newcommand{\DrawRubikCubeSidebarDBRD}{\side@barRubikBbackRD{1}{\Brb}\side@barRubikBbackRD{2}{\Bmb}\side@barRubikBbackRD{3}{\Blb}}

Now do the reverse (BD) = DB
\newcommand{\DrawRubikCubeSidebarBDRD}{\DrawRubikCubeSidebarDBRD}

Make the join commands
\newcommand{\DrawRubikCubeSidebarDB}[1]{\@join{\DrawRubikCubeSidebarDB}{#1}}\newcommand{\DrawRubikCubeSidebarBD}[1]{\@join{\DrawRubikCubeSidebarBD}{#1}}
20.17.4 Sidebars: LD view

Front-Up horizontal sidebar (LD view)

But FR (LD view) is the same as for (RU view), (see above)
\newcommand{\DrawRubikCubeSidebarFULD}{\DrawRubikCubeSidebarFURD}
\newcommand{\DrawRubikCubeSidebarUFLD}{\DrawRubikCubeSidebarUFRD}

Front-Right vertical sidebar (LD view)

Front Right (LD view) = same as for (LU view), (see above)
\newcommand{\DrawRubikCubeSidebarFRLD}{\DrawRubikCubeSidebarFRLU}
\newcommand{\DrawRubikCubeSidebarRFLD}{\DrawRubikCubeSidebarRFLU}

Left-Back vertical sidebar (LD view)

Modified from \side@barL (in Section 20.16.1) draws only a single small bar each of the three small bars has a numbered position (1,2,3); (dx,dy) = bottom Left corner of single facelet bar
\newcommand{\side@barRubikLbackLD}{\side@barRubikLbackLD}[2]{% 
\pgfmathsetmacro{\dx}{0 - \bs - \bw - 1} \ifthenelse{\equal{#2}{\no@sidebar}}{}{\pgfmathsetmacro{\dy}{#1 - 1 + 0.5 - \blh - 1} \pgfmathsetmacro{\blh}{\bl*(0.5)} \pgfmathsetmacro{\dx}{{\dx}} \pgfmathsetmacro{\dy}{{\dy}} \draw[fill=#2] (\dx,\dy) -- (\dx,\dy + \bl) -- (\dx + \bw, \dy + \bl) -- cycle; \}}}

Make the LB (LeftBack) version; bar 1 is at the bottom
\newcommand{\DrawRubikCubeSidebarLBLD}{\DrawRubikCubeSidebarLBLD}[2]{% 
\side@barRubikLbackLD{3}{\Brt} \side@barRubikLbackLD{2}{\Brm} \side@barRubikLbackLD{1}{\Brb} \}

Now do the reverse (BL) = LB
\newcommand{\DrawRubikCubeSidebarBLLD}{\DrawRubikCubeSidebarBLLD}[2]{% 
\DrawRubikCubeSidebarLBLD{\Blt} \DrawRubikCubeSidebarLBLD{\Blm} \DrawRubikCubeSidebarLBLD{\Blb} \}

Do NOT need to make the join commands (same as for the LU view)

Down-Back horizontal sidebar (LD view)

Modified from \side@barB (in Section 20.16.1) draws only a single small bar each of the three small bars has a numbered position (1,2,3); (dx,dy) = bottom Left corner of single facelet bar
The Rubik's cube is a 3x3x3 cube with different-colored faces. The goal is to solve the cube by rotating its faces so that each face becomes a single color.

The code provided is for drawing a Rubik's cube using TikZ, a package for producing graphics in LaTeX. The document explains how to draw a cube from different viewpoints and gives a history of the command's development.

The `\DrawNCubeAll` command takes four arguments:

- `#1` = number of cubies \((n>0)\) along each side,
- `#2`, `#3`, `#4` = colours of the visible faces (in X,Y,Z order); X=Right face colour, Y=Up face colour, Z=Front face colour.

The command uses the `\pgfmathsetmacro` command to calculate the positions of the cubies and their colors for each face.

The history of the command includes:

1. Adjusted to use the TikZ `\pgfmathsetmacro` command (suggested by Peter Grill).
2. Renamed to `\DrawNCubeAll`.

The `\DrawNCubeAll` command draws a solved NxNxN Rubik's cube from the RightUp viewpoint. All cubies on a given face have the same colour. The command takes four ordered arguments, as follows:

- `#1` = number of cubies \((n>0)\) along each side,
- `#2, #3, #4` = colours of the visible faces (in X,Y,Z order); X=Right face colour, Y=Up face colour, Z=Front face colour.

We use the `\pgfmathsetmacro{\text{variable-name}}{\text{numeric value or maths}}` command. Note that the second argument must not involve any units—just numeric values or mathematics.
This internal command is used only by the \DrawNCubeAll command (see above). The original version of this command was developed by Peter Bartal (see Bartal, 2011). It was later modified (2012) by RWD Nickalls (to implement a more intuitive X, Y, Z ordering of the face colour parameters).

The cube need not be in the origin, the distances of the down-behind [L] corner from the origin are taken as parameters 5, 6, 7. The command takes 7 ordered arguments:

1 - length of an edge
2 - X-face colour (RIGHT face)
3 - Y-face colour (UP face)
4 - Z-face colour (FRONT face)
5 - x-position in space
6 - y-position in space
7 - z-position in space

Usage:
\cube@dxdydz{1}{X}{Y}{Z}{x}{y}{z}

The original code \pgfmathparse{#1+#5}\let\dy\pgfmathresult was changed to the more intuitive \pgfmathsetmacro{dx}{#1+#5} (suggested by Peter Grill 2011).

Changes: RWD Nickalls (2012): (1) added the [line join=round, line cap=round] options to each of the TikZ \draw commands, in order to improve the line joining (first two options); (2) adjusted the \cube@dxdydz macro to adopt the ordered XYZ face colour notation (by reassigning #2, #3, #4 to the X, Y, Z face colours, as shown above).
20.19 Drawing single cubies

These two commands set the value of the two length parameters \(cx\) and \(cy\), and allow the user to vary the size (adjust \(cy\)) and horizontal viewpoint (adjust \(cx\)) of a single cubie (described in more detail in the RUBIKCUBE package documentation). Note that we cannot use the names \(dx\), \(dy\) for the variables here since these names have been allocated already (see above). However, we can use \(dx\), \(dy\) in the command names as these will be more readily understood by the user.

\newcommand{\Cubiedx}{1}{\pgfmathsetmacro{\cx}{#1}}
\newcommand{\Cubiedy}{1}{\pgfmathsetmacro{\cy}{#1}}

We now set the default values (\(cx=cy=0.4\))

\Cubiedx{0.4}
\Cubiedy{0.4}

These four commands draw a single cubie from the RightUp, RightDown, LeftUp, LeftDown viewpoint. The viewpoint is specified using an appended two-letter XY ordered viewpoint code: either RU, RD, LU, LD. These commands take three arguments, namely three different XYZ ordered colour codes (R,O,Y,G,B,W,X).

**FORMAT:** \texttt{\DrawCubieRU\{\textit{Xcolour}\}\{\textit{Ycolour}\}\{\textit{Zcolour}\}}

**USAGE:** \texttt{\DrawCubieRU\{G\}\{B\}\{W\}}
These four commands draw a single ‘text’ cubie from the RightUp, RightDown, LeftUp, LeftDown viewpoint. They are ‘text’ forms of the \DrawCubie commands described above. Their size was chosen to be suitable for use with 10–12 point fonts.

As before, the viewpoint is specified using an appended two-letter XY ordered viewpoint code: either RU, RD, LU, LD. These commands take three arguments (since just three faces are visible with this cube format), namely three different XYZ ordered colour codes (R,O,Y,G,B,W,X).

\text{FORMAT: } \textCubieRU{⟨Xcolour⟩}{⟨Ycolour⟩}{⟨Zcolour⟩}

\text{USAGE: } \textCubieRU{⟨Xcolour⟩}{⟨Ycolour⟩}{⟨Zcolour⟩}

\text{20.20 Text cubies}
20.21 Rotation commands

20.21.1 Introduction

We use a special prefix notation to denote each of four different representations of the various Rubik cube rotations as follows: the name of the Rubik rotation (rr), its associated hieroglyph (rrh), and combinations of name and hieroglyph both vertical (Rubik) and horizontal (textRubik). A rotation command is a combination of a rotation-code appended to one of the four prefixes.

For example, the command \rrhD generates the hieroglyph (rrh) associated with the rotation-code D. In this form it is used internally, but it is also available for the user.

In version 3.0, however, all the rotation commands were also made available to the user in the much more intuitive form stem{argument} form, for example, \rrh{D}. In practice, this ‘argument’ form actually generates the original non-argument form by the use of the internal macro \@join. For example, \rrh{D} \rightarrow \text{join(\rrh + D)} \rightarrow \rrhD (see Section 20.21.4 for details).

The hieroglyphs are of two types: ‘arrow’ glyphs (all exactly square), and ‘letter’ glyphs (mostly square, but many are rectangular); however both types are designed to have the same height so they sit nicely when arranged side-by-side. A lot of special macros for generating these glyphs are described below in Section 20.21.3 (and also in Section 20.21.2).

The ‘arrow’ hieroglyphs are built up in stages using TikZ. We first create a command for drawing the square (\DrawNotationBox; see Section 20.21.2) and
then draw the contents (lines, arrows, arcs of circles). For an example, see the
D form \( D \) constructed in Section 20.21.13.

The ‘letter’ hieroglyphs (glyphs for which the rotations cannot be seen from
the front, and hence cannot have arrows) just give a letter representation of the
rotation (say, Bw for ‘back wide’). These glyphs are therefore made using an fbox
(for convenience), and therefore these are sometimes not square. Some vertical
fine-tuning using the \( \texttt{raisebox} \) command is often required to force these ‘let-
ter’ glyphs to have the same vertical position as their ‘arrow’ cousins. A typical
example is the form \( Bw \) which is detailed in Section 20.21.7.

The presence of small overfilled \( \texttt{hbox} \) es associated with these squares were
originally checked for using the \( \texttt{ltugboat.cls} \), and all fixed mainly by setting
their associated small minipages \( \rightarrow \) width = 0.6cm, and using TikZ scale=0.5.

20.21.2 DrawNotationBox

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

This internal command draws the surrounding square box of all the hieroglyphs.

\[ \text{Note that we start at } (0,0) \text{ and draw to the final point in order to make a nice}
\text{ corner join.} \]

\text{TODO: ? make this a proper internal command using } \texttt{@} \text{ sometime.}

\begin{verbatim}
\newcommand{\DrawNotationBox}{%
\draw [thick] (0,0) -- (0,1) -- (1,1) -- (1,0) -- (0,0) -- (0,1)%
\}
\end{verbatim}

We now define a number of points and line-segments inside the square ‘nota-
tionbox’ (e.g., \( \@sd, \@sh \ldots \) etc.) which will be required for use in drawing the
various lines and arrows. Some hieroglyphs contain either one circular arc, or two
concentric arcs, and these arcs require both a centre and a start point. Note that
the final argument does not use any units. For the TikZ ARC command see TikZ

\text{TODO: make a small diagram to illustrate the position of these parameters and}
\text{ make things a bit clearer sometime.}

\begin{verbatim}
\pgfmathsetmacro{\@sd}{0.25} % a small horiz space
\pgfmathsetmacro{\@sdd}{2*\@sd} % 2x horiz space
\pgfmathsetmacro{\@sddd}{3*\@sd} % 3x horiz space
\pgfmathsetmacro{\@sh}{0.6} % height
\pgfmathsetmacro{\@sb}{0.2} % base
\pgfmathsetmacro{\@sbh}{\@sb + \@sh} % UP
\pgfmathsetmacro{\@scx}{\@sdd+0.13} % SQuare CenterX coord
\pgfmathsetmacro{\@sqcx}{\@scx-0.13} % Start of CircleX arc
\pgfmathsetmacro{\@scy}{\@sdd\@sd+0.2} % Start of CircleY arc
\pgfmathsetmacro{\@sqcy}{\@scy+0.25} % SQuare CenterY cpprd
\end{verbatim}

20.21.3 Some useful internal commands

\begin{verbatim}
\@rr
\@rrp
\@rrw
\@rrwp
\@rrs
\@rrsp
\@rra
\@rrap
\@xyzh
\@xyzhp
\@xyzRubik
\@xyzRubikp
\@SquareLetter
\end{verbatim}

These internal commands are used to generate the prime, w, w-prime, s, s-prime,
a, a-prime rotation commands. They attach a letter or a prime to the associated
argument; for example, the command \@rrwp{B} appends a ‘w’ and a prime (p) to the argument ‘B’, i.e. \[Bw\] (see Section 20.21.8). Users are then able to access this glyph by typing the command \rrw{Bwp}, or, more intuitively, \rr{Bwp} (see also \@join detailed in Section 20.21.4).

The \@xyz.. commands are used to generate the x, y, z, u, d, l, r, f, b commands and their associated prime rotation commands. The commands \@xyzhbdfl and \@xyzbdflRubik relate to the axis rotations denoted as b, d, f, l; since these four letters have long upstrokes they require special fine-tuning for vertical position.

The \@SquareLetter command is used to form the separate square hieroglyph form used for rotations with no visible representation from the front (e.g., B., Fs, Fsp, Fa, Fap, S, Sp, Sf, Sfp, Sh, Sbp). Note that the TikZ ‘thick’ line code = 0.8pt (used in \@SquareLetter). The \@hRubik is the vertical shift used to raise the box carrying the rotation rotation-code in \Rubik.. commands not visible from the front.

The idea is that by using these internal tools taking parameters we are able to more easily standardise the size and position of all the various glyphs. For details of the rubikfont and rubikprime see Section 20.3).

\begin{verbatim}
1134 \newcommand{\@rr}[1]{{\rubikfont #1}}
1135 \newcommand{\@rrp}[1]{{\rubikfont #1\rubikprime}}
1136 \newcommand{\@rrwp}[1]{{\rubikfont #1\rubikfontFNS w}}
1137 \newcommand{\@rrwp}[1]{{\rubikfont #1\rubikfontFNS w\rubikprime}}
1138 \newcommand{\@rrs}[1]{{\rubikfont #1\rubikfontFNS s}}
1139 \newcommand{\@rrsp}[1]{{\rubikfont #1\rubikfontFNS s\rubikprime}}
1140 \newcommand{\@rra}[1]{{\rubikfont #1\rubikfontFNS a}}
1141 \newcommand{\@rrap}[1]{{\rubikfont #1\rubikfontFNS a\rubikprime}}
1142 \newcommand{\@rru}[1]{{\rubikfont #1\rubikfontFNS u}}
1143 \newcommand{\@rrup}[1]{{\rubikfont #1\rubikfontFNS u\rubikprime}}
1144 \newcommand{\@rrd}[1]{{\rubikfont #1\rubikfontFNS d}}
1145 \newcommand{\@rrdp}[1]{{\rubikfont #1\rubikfontFNS d\rubikprime}}
1146 \newcommand{\@rrl}[1]{{\rubikfont #1\rubikfontFNS l}}
1147 \newcommand{\@rrlp}[1]{{\rubikfont #1\rubikfontFNS l\rubikprime}}
1148 \newcommand{\@rrr}[1]{{\rubikfont #1\rubikfontFNS r}}
1149 \newcommand{\@rrrp}[1]{{\rubikfont #1\rubikfontFNS r\rubikprime}}
1150 \newcommand{\@rrf}[1]{{\rubikfont #1\rubikfontFNS f}}
1151 \newcommand{\@rrfp}[1]{{\rubikfont #1\rubikfontFNS f\rubikprime}}
1152 \newcommand{\@rrb}[1]{{\rubikfont #1\rubikfontFNS b}}
1153 \newcommand{\@rrbp}[1]{{\rubikfont #1\rubikfontFNS b\rubikprime}}
1154 \newcommand{\@rrc}[1]{{\rubikfont #1\rubikfontFNS c}}
1155 \newcommand{\@rrcp}[1]{{\rubikfont #1\rubikfontFNS c\rubikprime}}
1156 \newcommand{\@rrm}[1]{{\rubikfont #1\rubikfontFNS m}}
1157 \newcommand{\@rrmp}[1]{{\rubikfont #1\rubikfontFNS m\rubikprime}}
1158 \newcommand{\@xyz}[1]{{\rubikfont #1}}
1159 \newcommand{\@xyzhp}[1]{{\rubikfont #1\raisebox{0.6pt}{\rubikprime}}}
1160 \newcommand{\@xyzRubik}[1]{{\rubikfont #1}}
1161 \{\raisebox{3.45pt}{{\rubikfont #1}}\}
1162 \newcommand{\@xyzRubik}[1]{{\rubikfont #1}}
1163 \{\raisebox{3.45pt}{{\rubikfont #1}}\}
\end{verbatim}
Feb 2017 (RWDN): We also need to define a small length for fine-tuning the default horizontal space between a pair of ‘letter’ hieroglyphs, eg B (i.e., when no additional space has been added by the user), so that this matches that between a pair of ‘arrow’ hieroglyphs. This length is inserted on both sides of the square frame. This length is used in two settings: (a) in ‘letter’ hieroglyphs (for an example, see the definition of the macro \SquareB in Section 20.21.5), and in (b) in ‘arrow’ hieroglyphs (for an example, see the definition of the macro \rrhD in Section 20.21.13).

\textRubik

\Rubik

\rrh

\rrrh

The following four commands typeset a single rotation, where a rotation-code (e.g., U) is the argument (see Section 9.1). As an example, the format for the \rrh{} command is \texttt{\rrh{\{rotation-code\}}}. In practice, these four commands are really a sort of front-end for all the commands which follow this section. For example, the command \texttt{\rrh{U}} generates the command \texttt{\rrhU} which itself typesets the rotation hieroglyph for the rotation U, etc.

These four commands, which use the internal \texttt{\@join} command (see above), are especially useful when typesetting a list of rotation-codes. Furthermore, it is
more intuitive for the user to specify a rotation command using the rotation-code as an argument.

\newcommand{\Rubik}[1]{\@join{\Rubik}{#1}}
\newcommand{\textRubik}[1]{\@join{\textRubik}{#1}}
\newcommand{\rr}{\@join{\rr}{#1}}
\newcommand{\rrh}{\@join{\rrh}{#1}}

20.21.5 Rotation B

These commands all draw forms which denote the B (back-face) rotation. Not visible from the front.

Feb 2017 (RWDN): added the \@tlen length (= 1pt; defined above) to the \SquareB command, and removed the terminal \, space from the rrB, RubikB, textRubikB commands, and copied this action with all the subsequent Letter hieroglyphs (e.g., B, Bw, ...). These minor changes were to improve the spacing between two Letter hieroglyphs, and make it match that between two square ‘arrow’ hieroglyphs. The same changes were made to all the ‘letter’ hieroglyphs.

\newcommand{\rrB}{\@rr{B}}
\newcommand{\SquareB}{\@tlen\@SquareLetter{\rrB}\@tlen}
\newcommand{\rrhB}{\raisebox{-0.25mm}{\SquareB}}
\newcommand{\RubikB}{\raisebox{\@hRubik}{\SquareB}}
\newcommand{\textRubikB}{\rrhB}

20.21.6 Rotation Bp

These commands all draw forms which denote the Bp rotation. Not visible from the front.

\newcommand{\rrpB}{\@rrp{B}}
\newcommand{\SquareBp}{\@tlen\@SquareLetter{\rrpB}\@tlen}
\newcommand{\rrhpB}{\raisebox{-0.25mm}{\SquareBp}}
\newcommand{\RubikBp}{\raisebox{\@hRubik}{\SquareBp}}
\newcommand{\textRubikBp}{\rrhpB}

20.21.7 Rotation Bw

These commands all draw forms which denote the Bw rotation. Not visible from the front.

\newcommand{\rrwB}{\@rrw{B}}
\newcommand{\SquareBw}{\@tlen\@SquareLetter{\rrwB}\@tlen}
\newcommand{\rrhwB}{\raisebox{-0.25mm}{\SquareBw}}
\newcommand{\RubikBw}{\raisebox{\@hRubik}{\SquareBw}}
\newcommand{\textRubikBw}{\rrhwB}

20.21.8 Rotation Bwp

These commands all draw forms which denote the Bwp rotation. Not visible from the front.
20.21.9 Rotation Bs

These commands all draw forms which denote the Bs rotation. Not visible from the front.

\newcommand{\rrBs}{\@rrs{B}}
\newcommand{\SquareBs}{\@tlen\@SquareLetter{\rrBs}\@tlen}
\newcommand{\rrhBs}{\raisebox{-0.25mm}{\SquareBs}}
\newcommand{\RubikBs}{\raisebox{\@hRubik}{\SquareBs}}
\newcommand{\textRubikBs}{\rrhBs}

20.21.10 Rotation Bsp

These commands all draw forms which denote the Bsp rotation. Not visible from the front.

\newcommand{\rrBsp}{\@rrsp{B}}
\newcommand{\SquareBsp}{\@tlen\@SquareLetter{\rrBsp}\@tlen}
\newcommand{\rrhBsp}{\raisebox{-0.25mm}{\SquareBsp}}
\newcommand{\RubikBsp}{\raisebox{\@hRubik}{\SquareBsp}}
\newcommand{\textRubikBsp}{\rrhBsp}

20.21.11 Rotation Ba

These commands all draw forms which denote the Ba rotation. Not visible from the front.

\newcommand{\rrBa}{\@rra{B}}
\newcommand{\SquareBa}{\@tlen\@SquareLetter{\rrBa}\@tlen}
\newcommand{\rrhBa}{\raisebox{-0.25mm}{\SquareBa}}
\newcommand{\RubikBa}{\raisebox{\@hRubik}{\SquareBa}}
\newcommand{\textRubikBa}{\rrhBa}

20.21.12 Rotation Bap

These commands all draw forms which denote the Bap rotation. Not visible from the front.

\newcommand{\rrBap}{\@rrap{B}}
\newcommand{\SquareBap}{\@tlen\@SquareLetter{\rrBap}\@tlen}
\newcommand{\rrhBap}{\raisebox{-0.25mm}{\SquareBap}}
\newcommand{\RubikBap}{\raisebox{\@hRubik}{\SquareBap}}
\newcommand{\textRubikBap}{\rrhBap}
20.21.13 Rotation D

\rrD These commands all draw forms which denote the D rotation.

Feb 2017 (RWDN): added the \@tlen length to the \rrhD command to improve the spacing between two ‘arrow’ square hieroglyphs; and also removed the terminal \, space. The same changes were made to all the ‘arrow’ hieroglyphs.

1224 \newcommand{\rrD}{\@rr{D}}
1225 \%
1226 \newcommand{\SquareD}{%
1227 \begin{tikzpicture}[scale=0.5]
1228 \DrawNotationBox;
1229 \draw [thick] (\@sb,\@sddd) -- (\@sbh, \@sddd);
1230 \draw [thick] (\@sb,\@sdd) -- (\@sbh, \@sdd);
1231 \draw [thick, ->] (\@sb,\@sd) -- (\@sbh, \@sd);
1232 \end{tikzpicture}%
1233 }
1234 \newcommand{\rrhD}{\raisebox{-0.333\height}{\@tlen\SquareD\@tlen}}
1235 \%
1236 \newcommand{\RubikD}{%
1237 {\@rubikfont
1238 \begin{minipage}{0.6cm}
1239 \centering
1240 \SquareD\rrD
1241 \end{minipage}%
1242 }
1243 \newcommand{\textRubikD}{\rrD\,\rrhD}

20.21.14 Rotation Dp

\rrDp These commands all draw forms which denote the Dp rotation.

1245 \newcommand{\rrDp}{\@rrp{D}}
1246 \%
1247 \newcommand{\SquareDp}{%
1248 \begin{tikzpicture}[scale=0.5]
1249 \DrawNotationBox;
1250 \draw [thick] (\@sb,\@sddd) -- (\@sbh, \@sddd);
1251 \draw [thick] (\@sb,\@sdd) -- (\@sbh, \@sdd);
1252 \draw [thick, ->] (\@sb,\@sd) -- (\@sbh, \@sd);
1253 \end{tikzpicture}%
1254 }
1255 \newcommand{\rrhDp}{\raisebox{-0.333\height}{\@tlen\SquareDp\@tlen}}
1256 \%
1257 \newcommand{\RubikDp}{%
1258 {\@rubikfont
1259 \begin{minipage}{0.6cm}
1260 \centering
1261 \SquareDp\rrDp
1262 \end{minipage}%;
20.21.15 Rotation Dw

These commands all draw forms which denote the Dw rotation.

\newcommand{\rrDw}{\@rrw{D}}
\newcommand{\SquareDw}{\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\@sb,\@sddd) -- (\@sbh, \@sddd);
\draw [thick, ->] (\@sb,\@sdd) -- (\@sbh, \@sdd);
\draw [thick, ->] (\@sb,\@sd) -- (\@sbh, \@sd);
\end{tikzpicture}}
\newcommand{\rrhDw}{\raisebox{-0.333\height}{\@tlen\SquareDw\@tlen}}
\newcommand{\RubikDw}{\@rubikfont\begin{minipage}{0.6cm}
\centering\SquareDw\rrDw\end{minipage}}
\newcommand{\textRubikDw}{\rrDw,\rrhDw}

20.21.16 Rotation Dwp

These commands all draw forms which denote the Dwp rotation.

\newcommand{\rrDwp}{\@rrwp{D}}
\newcommand{\SquareDwp}{\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\@sb,\@sddd) -- (\@sbh, \@sddd);
\draw [thick, ->] (\@sb,\@sdd) -- (\@sbh, \@sdd);
\draw [thick, <-] (\@sb,\@sd) -- (\@sbh, \@sd);
\end{tikzpicture}}
\newcommand{\rrhDwp}{\raisebox{-0.333\height}{\@tlen\SquareDwp\@tlen}}
\newcommand{\RubikDwp}{\@rubikfont\begin{minipage}{0.6cm}
\centering\SquareDwp\rrDwp\end{minipage}}
\newcommand{\textRubikDwp}{\rrDwp,\rrhDwp}
\textRubikDsp} \end{minipage}}}

\newcommand{\textRubikDs}{\rrDs, \rrhDs}

\section*{20.21.17 Rotation Ds}

These commands all draw forms which denote the Ds rotation.

\newcommand{\rrDs}{\@rrs{D}}

\newcommand{\SquareDs}{\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\@sb, \@sddd) -- (\@sbh, \@sddd);
\draw [thick] (\@sb, \@sdd) -- (\@sbh, \@sdd);
\draw [thick, ->] (\@sb, \@sd) -- (\@sbh, \@sd);
\end{tikzpicture}}

\newcommand{\rrhDs}{\raisebox{-0.333\height}{\@tlen\SquareDs\@tlen}}

\newcommand{\RubikDs}{\@rubikfont\begin{minipage}{0.6cm}
\centering\SquareDs\rrDs\end{minipage}}

\newcommand{\textRubikDs}{\rrDs, \rrhDs}

\section*{20.21.18 Rotation Dsp}

These commands all draw forms which denote the Dsp rotation.

\newcommand{\rrDsp}{\@rrsp{D}}

\newcommand{\SquareDsp}{\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\@sb, \@sddd) -- (\@sbh, \@sddd);
\draw [thick] (\@sb, \@sdd) -- (\@sbh, \@sdd);
\draw [thick, ->] (\@sb, \@sd) -- (\@sbh, \@sd);
\end{tikzpicture}}

\newcommand{\rrhDsp}{\raisebox{-0.333\height}{\@tlen\SquareDsp\@tlen}}

\newcommand{\RubikDsp}{\@rubikfont\begin{minipage}{0.6cm}
\centering
\newcommand{\textRubikDsp}{\rrDsp \rrhDsp}

20.21.19 Rotation Da

These commands all draw forms which denote the Da rotation.
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\@sb,\@sddd) -- (\@sbh, \@sddd);
\draw [thick] (\@sb,\@sdd) -- (\@sbh, \@sdd);
\draw [thick, <-] (\@sb,\@sd) -- (\@sbh, \@sd);
\end{tikzpicture}
\newcommand{\textRubikDa}{\rrDa \rrhDa}

20.21.20 Rotation Dap

These commands all draw forms which denote the Dap rotation.
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\@sb,\@sddd) -- (\@sbh, \@sddd);
\draw [thick] (\@sb,\@sdd) -- (\@sbh, \@sdd);
\draw [thick, <-] (\@sb,\@sd) -- (\@sbh, \@sd);
\end{tikzpicture}
20.21.21 Rotation E

These commands all draw forms which denote the E rotation.

\newcommand{\rrE}{\@rr{E}}
\newcommand{\SquareE}{\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\@sb,\@sddd) -- (\@sbh, \@sddd);
\draw [thick, ->] (\@sb, \@sdd) -- (\@sbh, \@sdd);
\draw [thick] (\@sb, \@sd) -- (\@sbh, \@sd);
\end{tikzpicture}}
\newcommand{\rrhE}{\raisebox{-0.333\height}{\@tlen \SquareE \@tlen}}
\newcommand{\RubikE}{\@rubikfont\begin{minipage}{0.6cm}
\centering \SquareE \rrE\end{minipage}}
\newcommand{\textRubikE}{\rrE \rrhE}

20.21.22 Rotation Ep

These commands all draw forms which denote the Ep rotation.

\newcommand{\rrEp}{\@rrp{E}}
\newcommand{\SquareEp}{\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\@sb,\@sddd) -- (\@sbh, \@sddd);
\draw [thick, <-] (\@sb, \@sdd) -- (\@sbh, \@sdd);
\draw [thick] (\@sb, \@sd) -- (\@sbh, \@sd);
\end{tikzpicture}}
\newcommand{\rrhEp}{\raisebox{-0.333\height}{\@tlen \SquareEp \@tlen}}
\newcommand{\RubikEp}{\@rubikfont\begin{minipage}{0.6cm}
\centering \SquareEp \rrEp\end{minipage}}
\newcommand{\textRubikEp}{\rrEp \rrhEp}
These commands all draw forms which denote the F rotation.

\begin{minipage}{0.6cm}
\centering
\SquareEp\\rrEp
\end{minipage}
\newcommand{\textRubikEp}{\rrEp,\rrhEp}

\section{Rotation F}

These commands all draw forms which denote the Fp rotation.

\begin{minipage}{0.6cm}
\centering
\SquareFp\rrFP
\end{minipage}
\newcommand{\textRubikFp}{\rrFP,\rrhFP}
These commands all draw forms which denote the Fw rotation.

\newcommand{\rrFw}{\@rrw{F}}
\newcommand{\SquareFw}{\begin{tikzpicture}\[scale=0.5\]
\DrawNotationBox;
\draw[thick,->] (\@scx, \@scy) arc[radius=0.35, start angle=-60, delta angle=290];
\draw[thick] (\@scx, \@scy) arc[radius=0.1, start angle=-60, delta angle=360];
%\node (squareLab) at (0.5,0.5) {$o$};
\end{tikzpicture}}
\newcommand{\rrhFw}{\raisebox{-0.333\height}{\@tlen\SquareFw\@tlen}}
\newcommand{\RubikFw}{\begin{minipage}{0.6cm}
\centering\SquareFw\rrFw\end{minipage}}

These commands all draw forms which denote the Fwp rotation.

\newcommand{\rrFwp}{\@rrwp{F}}
\newcommand{\SquareFwp}{\begin{tikzpicture}\[scale=0.5\]
\DrawNotationBox;
\draw[thick,->] (\@scx, \@scy) arc[radius=0.35, start angle=-60, delta angle=290];
\draw[thick] (\@scx, \@scy) arc[radius=0.1, start angle=-60, delta angle=360];
%\node (squareLab) at (0.5,0.5) {$o$};
\end{tikzpicture}}
\newcommand{\rrhFwp}{\raisebox{-0.333\height}{\@tlen\SquareFwp\@tlen}}
\newcommand{\RubikFwp}{\begin{minipage}{0.6cm}
\centering\SquareFwp\rrFwp\end{minipage}}
These commands draw forms of the Singmaster Fs slice rotation. We need to just make square with Fs in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by \{} so no need to reset to defaults. Not visible from the front.

\newcommand{\textRubikFs}{\rrhFs}

These commands draw forms of the Singmaster Fsp slice rotation. We need to just make square with Fsp in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by \{} so no need to reset to defaults. Not visible from the front.

\newcommand{\textRubikFsp}{\rrhFsp}

These commands draw forms of the Singmaster Fa slice rotation. We need to just make square with Fa in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by \{} so no need to reset to defaults. Not visible from the front.

\newcommand{\textRubikFa}{\rrhFa}

These commands draw forms of the Singmaster Fap slice rotation. We need to just make square with Fap in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by \{} so no need to reset to defaults. Not visible from the front.

\newcommand{\textRubikFap}{\rrhFap}
These commands all draw forms which denote the L rotation.

20.21.32 Rotation Lp

These commands all draw forms which denote the Lp rotation.
20.21.33 Rotation Lw

These commands all draw forms which denote the Lw rotation.

\newcommand{\rrLw}{\@rrw{L}}
\newcommand{\SquareLw}{\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\@sd, \@sb) -- (\@sd, \@sbh);
\draw [thick, ->] (\@sdd, \@sb) -- (\@sdd, \@sbh);
\draw [thick] (\@sddd, \@sb) -- (\@sddd, \@sbh);
\end{tikzpicture}}
\newcommand{\rrhLw}{\raisebox{-0.333\height}{\@tlen\SquareLw\@tlen}}
\newcommand{\RubikLw}{\@rubikfont\begin{minipage}{0.6cm}
\centering\SquareLw\rrLw\end{minipage}}
\newcommand{\textRubikLw}{\rrLw,\rrhLw}

20.21.34 Rotation Lwp

These commands all draw forms which denote the Lwp rotation.

\newcommand{\rrLwp}{\@rrwp{L}}
\newcommand{\SquareLwp}{\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\@sd, \@sb) -- (\@sd, \@sbh);
\draw [thick, ->] (\@sdd, \@sb) -- (\@sdd, \@sbh);
\draw [thick] (\@sddd, \@sb) -- (\@sddd, \@sbh);
\end{tikzpicture}}
\newcommand{\rrhLwp}{\raisebox{-0.333\height}{\@tlen\SquareLwp\@tlen}}
\newcommand{\RubikLwp}{\@rubikfont\begin{minipage}{0.6cm}
\centering\SquareLwp\rrLwp\end{minipage}}
\newcommand{\textRubikLwp}{\rrLwp,\rrhLwp}
\begin{minipage}{0.6cm}
centering
\SquareLwp\
\rrLwp
\end{minipage}
}
\newcommand{\textRubikLwp}{\rrLwp,\rrhLwp}

\newcommand{\textRubikLs}{\rrLs,\rrhLs}

These commands all draw forms which denote the Ls rotation.

\newcommand{\rrLsp}{\@rrsp{L}}

\newcommand{\SquareLsp}{\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\@sd, \@sb) -- (\@sd, \@sbh);
\draw [thick] (\@sdd,\@sb) -- (\@sdd, \@sbh);
\draw [thick, ->] (\@sddd, \@sb) -- (\@sddd, \@sbh);
\end{tikzpicture}}

\newcommand{\textRubikLsp}{\rrLsp,\rrhLsp}

These commands all draw forms which denote the Lsp rotation.
\newcommand{\textRubikLa}{\rrLa\textRubikLsp}

These commands all draw forms which denote the La rotation.

\newcommand{\rrLa}{\@rra{L}}
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\@sd, \@sb) -- (\@sd, \@sbh);
\draw [thick] (\@sdd,\@sb) -- (\@sdd, \@sbh);
\draw [thick, <-] (\@sddd, \@sb) -- (\@sddd, \@sbh);
\end{tikzpicture}
\newcommand{\rrhLa}{\raisebox{-0.333\height}{\@tlen\SquareLa\@tlen}}
\begin{minipage}{0.6cm}
\centering
\SquareLa\rrLa
\end{minipage}
\newcommand{\RubikLa}{\@rubikfont\begin{minipage}{0.6cm}
\centering
\SquareLa\rrLa
\end{minipage}}
\newcommand{\textRubikLa}{\rrLa\textRubikLsp}

20.21.37 Rotation La

These commands all draw forms which denote the La rotation.

\newcommand{\rrLap}{\@rrap{L}}
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\@sd, \@sb) -- (\@sd, \@sbh);
\draw [thick] (\@sdd,\@sb) -- (\@sdd, \@sbh);
\draw [thick, <-] (\@sddd, \@sb) -- (\@sddd, \@sbh);
\end{tikzpicture}
\newcommand{\rrhLap}{\raisebox{-0.333\height}{\@tlen\SquareLap\@tlen}}
\begin{minipage}{0.6cm}
\centering
\SquareLap\rrhLap
\end{minipage}
\newcommand{\RubikLap}{\@rubikfont\begin{minipage}{0.6cm}
\centering
\SquareLap\rrhLap
\end{minipage}}
\newcommand{\textRubikLap}{\rrhLsp\textRubikLa}

20.21.38 Rotation Lap

These commands all draw forms which denote the Lap rotation.

\newcommand{\rrLap}{\@rrap{L}}
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\@sd, \@sb) -- (\@sd, \@sbh);
\draw [thick] (\@sdd,\@sb) -- (\@sdd, \@sbh);
\draw [thick, <-] (\@sddd, \@sb) -- (\@sddd, \@sbh);
\end{tikzpicture}
\newcommand{\rrhLap}{\raisebox{-0.333\height}{\@tlen\SquareLap\@tlen}}
\begin{minipage}{0.6cm}
\centering
\SquareLap\rrhLap
\end{minipage}
\newcommand{\RubikLap}{\@rubikfont\begin{minipage}{0.6cm}
\centering
\SquareLap\rrhLap
\end{minipage}}
\newcommand{\textRubikLap}{\rrhLsp\textRubikLa}
These commands all draw forms which denote the M rotation.

20.21.40 Rotation Mp

These commands all draw forms which denote the Mp rotation.
20.21.41 Rotation R

These commands all draw forms which denote the R rotation.

\text{Rotation R}
\newcommand{\RR}{\@rr{R}}
\newcommand{\SquareR}{\\begin{tikzpicture}\[scale=0.5\]}
\draw [thick] (\@sd, \@sb) -- (\@sd, \@sbh);
\draw [thick] (\@sdd,\@sb) -- (\@sdd, \@sbh);
\draw [thick, ->] (\@sddd, \@sb) -- (\@sddd, \@sbh);
\end{tikzpicture}}
\newcommand{\RRh}{\raisebox{-0.333\height}{\@tlen\SquareR\@tlen}}
\newcommand{\RubikR}{\@rubikfont\begin{minipage}{0.6cm}\centering\SquareR\rrR\end{minipage}}
\newcommand{\textRubikR}{\rrR,\rrhR}

20.21.42 Rotation Rp

These commands all draw forms which denote the Rp rotation.

\text{Rotation Rp}
\newcommand{\RRp}{\@rrp{R}}
\newcommand{\SquareRp}{\\begin{tikzpicture}\[scale=0.5\]}
\draw [thick] (\@sd, \@sb) -- (\@sd, \@sbh);
\draw [thick] (\@sdd,\@sb) -- (\@sdd, \@sbh);
\draw [thick, <-] (\@sddd, \@sb) -- (\@sddd, \@sbh);
\end{tikzpicture}}
20.21.43 Rotation Rw

These commands all draw forms which denote the Rw rotation.

\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\@sd, \@sb) -- (\@sd, \@sbh);
\draw [thick, ->] (\@sdd, \@sb) -- (\@sdd, \@sbh);
\draw [thick, ->] (\@sddd, \@sb) -- (\@sddd, \@sbh);
\end{tikzpicture}

20.21.44 Rotation Rwp

These commands all draw forms which denote the Rwp rotation.

\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\@sd, \@sb) -- (\@sd, \@sbh);
\draw [thick, <-] (\@sdd, \@sb) -- (\@sdd, \@sbh);
\draw [thick, <-] (\@sddd, \@sb) -- (\@sddd, \@sbh);
\begin{tikzpicture}
\end{tikzpicture}

%%

\newcommand{\rrhRs}{\raisebox{-0.333\height}{\@tlen\SquareRs\@tlen}}
%%
\newcommand{\RubikRs}{\@rubikfont\begin{minipage}{0.6cm}\centering\SquareRs\rrRs\end{minipage}}
\newcommand{\textRubikRs}{\rrRs\rrhRs}

\newcommand{\rrRs}{\@rrs{R}}
%%
\newcommand{\SquareRs}{\begin{tikzpicture}[scale=0.5]\DrawNotationBox;\draw [thick,->] (\@sd, \@sb) -- (\@sd, \@sbh);\draw [thick] (\@sdd,\@sb) -- (\@sdd, \@sbh);\draw [thick,->] (\@sddd, \@sb) -- (\@sddd, \@sbh);\end{tikzpicture}}
\newcommand{\rrhRs}{\raisebox{-0.333\height}{\@tlen\SquareRs\@tlen}}
%%
\newcommand{\RubikRs}{\@rubikfont\begin{minipage}{0.6cm}\centering\SquareRs\rrRs\end{minipage}}
\newcommand{\textRubikRs}{\rrRs\rrhRs}

\newcommand{\rrRsp}{\@rrsp{R}}
%%
\newcommand{\SquareRsp}{\begin{tikzpicture}[scale=0.5]\DrawNotationBox;\draw [thick,<-] (\@sd, \@sb) -- (\@sd, \@sbh);\draw [thick] (\@sdd,\@sb) -- (\@sdd, \@sbh);\draw [thick,<-] (\@sddd, \@sb) -- (\@sddd, \@sbh);\end{tikzpicture}}
\newcommand{\rrhRsp}{\raisebox{-0.333\height}{\@tlen\SquareRsp\@tlen}}
%%
\newcommand{\RubikRsp}{\@rubikfont\begin{minipage}{0.6cm}\centering\SquareRsp\rrRsp\end{minipage}}
\newcommand{\textRubikRsp}{\rrRsp\rrhRsp}

\newcommand{\rrRsp}{\@rrsp{R}}
%%
\newcommand{\SquareRsp}{\begin{tikzpicture}[scale=0.5]\DrawNotationBox;\draw [thick,<-] (\@sd, \@sb) -- (\@sd, \@sbh);\draw [thick] (\@sdd,\@sb) -- (\@sdd, \@sbh);\draw [thick,<-] (\@sddd, \@sb) -- (\@sddd, \@sbh);\end{tikzpicture}}
These commands all draw forms which denote the Ra rotation.

\newcommand{\rrRa}{\@rra{R}}
\newcommand{\SquareRa}{\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick,->] (\@sd, \@sb) -- (\@sd, \@sbh);
\draw [thick] (\@sdd,\@sb) -- (\@sdd, \@sbh);
\draw [thick,<->] (\@sddd, \@sb) -- (\@sddd, \@sbh);
\end{tikzpicture}}
\newcommand{\rrhRa}{\raisebox{-0.333\height}{\@tlen\SquareRa\@tlen}}
\newcommand{\RubikRa}{\@rubikfont\begin{minipage}{0.6cm}
\centering\SquareRa\rrRa\end{minipage}}
\newcommand{\textRubikRa}{\rrRa\,\rrhRa}

These commands all draw forms which denote the Rap rotation.

\newcommand{\rrRap}{\@rrap{R}}
\newcommand{\SquareRap}{\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick,->] (\@sd, \@sb) -- (\@sd, \@sbh);
\draw [thick] (\@sdd,\@sb) -- (\@sdd, \@sbh);
\draw [thick,<->] (\@sddd, \@sb) -- (\@sddd, \@sbh);
\end{tikzpicture}}
\newcommand{\rrhRap}{\raisebox{-0.333\height}{\@tlen\SquareRap\@tlen}}
\newcommand{\RubikRap}{\@rubikfont\begin{minipage}{0.6cm}
\centering\SquareRap\rrRap\end{minipage}}
\newcommand{\textRubikRap}{\rrRap\,\rrhRap}
\newcommand{\rrhRap}{\raisebox{-0.333\height}{\@tlen\SquareRap\@tlen}}
\newcommand{\RubikRap}{\begin{minipage}{0.6cm}
\centering
\SquareRap
\rrRap
\end{minipage}}
\newcommand{\textRubikRap}{\rrhRap\,\rrhRap}

20.21.49 Rotation S
\rrS These commands all draw forms which denote the S rotation. Not visible from the front.
\SquareS \rrhS \RubikS \textRubikS
\newcommand{\rrS}{\@rr{S}}
\newcommand{\SquareS}{\@tlen\@SquareLetter{\rrS}\@tlen}
\newcommand{\rrhS}{\raisebox{-0.25mm}{\SquareS}}
\newcommand{\RubikS}{\raisebox{\@hRubik}{\SquareS}}
\newcommand{\textRubikS}{\rrhS}

20.21.50 Rotation Sp
\rrSp These commands all draw forms which denote the Sp rotation. Not visible from the front.
\SquareSp \rrhSp \RubikSp \textRubikSp
\newcommand{\rrSp}{\@rrp{S}}
\newcommand{\SquareSp}{\@tlen\@SquareLetter{\rrSp}\@tlen}
\newcommand{\rrhSp}{\raisebox{-0.25mm}{\SquareSp}}
\newcommand{\RubikSp}{\raisebox{\@hRubik}{\SquareSp}}
\newcommand{\textRubikSp}{\rrhSp}

20.21.51 Rotation Su
These commands draw forms of the Singmaster Su slice rotation. We also need to fine-tune the spacing between these ‘slice’ hieroglyphs (especially Fs and Bs).
\rrSu \rrhSu \RubikSu \textRubikSu
\newcommand{\rrSu}{\@rru{S}}
\newcommand{\rrhSu}{\rrhEp}\textRubikSu
\newcommand{\RubikSu}{\begin{minipage}{0.6cm}
\centering
\SquareEp
\rrSu
\end{minipage}}
\newcommand{\textRubikSu}{\rrhSu}
\newcommand{\rrhSu}{\@rru{S}}
\newcommand{\RubikSu}{\begin{minipage}{0.6cm}
\centering
\SquareEp
\rrhSu
\end{minipage}}
\newcommand{\textRubikSu}{\rrhSu}
\newcommand{\sqrt}{\@sqrt\sqrt}\textRubikSu
\newcommand{\RubikSu}{\begin{minipage}{0.6cm}
\centering
\SquareEp
\rrhSu
\end{minipage}}
\newcommand{\textRubikSu}{\rrhSu}
\newcommand{\sqrt}{\@sqrt\sqrt}\textRubikSu
\newcommand{\RubikSu}{\begin{minipage}{0.6cm}
\centering
\SquareEp
\rrhSu
\end{minipage}}
\newcommand{\textRubikSu}{\rrhSu}
These commands draw forms of the Singmaster Sup slice rotation. We also need to fine-tune the spacing between these ‘slice’ hieroglyphs (especially Fs and Bs).

\newcommand{\textRubikSu}{\rrSu,\rrhEp}

20.21.52 Rotation Sup

These commands draw forms of the Singmaster Sup slice rotation.

\newcommand{\rrSup}{\@rrup{S}}
\newcommand{\rrhSup}{\rrhE}\
\newcommand{\RubikSup}{\@rubikfont\begin{minipage}{0.6cm}\centering\SquareE\rrSup\end{minipage}}
\newcommand{\textRubikSup}{\rrSup,\rrhE}

20.21.53 Rotation Sd

These commands draw forms of the Singmaster Sd slice rotation.

\newcommand{\rrSd}{\@rrd{S}}
\newcommand{\rrhSd}{\rrhE}\
\newcommand{\RubikSd}{\@rubikfont\begin{minipage}{0.6cm}\centering\SquareE\rrSd\end{minipage}}
\newcommand{\textRubikSd}{\rrSd,\rrhE}

20.21.54 Rotation Sdp

These commands draw forms of the Singmaster Sdp slice rotation.

\newcommand{\rrSdp}{\@rrdp{S}}
\newcommand{\rrhSdp}{\rrhEp}%
\newcommand{\RubikSdp}{\@rubikfont\begin{minipage}{0.6cm}\centering\SquareEp\rrSdp\end{minipage}}
\newcommand{\textRubikSdp}{\rrSdp,\rrhEp}
20.21.55 Rotation Sl

These commands draw forms of the Singmaster Sl slice rotation.

\newcommand{\rrSl}{\@rrl{S}}\hfill 1967
\newcommand{\rrhSl}{\rrhM} 1968
\newcommand{\RubikSl}{% \@rubikfont% \begin{minipage}{0.6cm} \centering\SquareM\ \rrSl\end{minipage} %} \hfill \newcommand{\textRubikSl}{\rrSl\ \rrhM} \hfill 1969

20.21.56 Rotation Slp

These commands draw forms of the Singmaster Slp slice rotation.

\newcommand{\rrSlp}{\@rrlp{S}}\hfill 1978
\newcommand{\rrhSlp}{\rrhMp} 1979
\newcommand{\RubikSlp}{% \@rubikfont% \begin{minipage}{0.6cm} \centering\SquareMp\ \rrSlp\end{minipage} %} \hfill \newcommand{\textRubikSlp}{\rrSlp\ \rrhMp} \hfill 1980

20.21.57 Rotation Sr

These commands draw forms of the Singmaster Sr slice rotation.

\newcommand{\rrSr}{\@rrr{S}}\hfill 1989
\newcommand{\rrhSr}{\rrhMp} 1990
\newcommand{\RubikSr}{% \@rubikfont% \begin{minipage}{0.6cm} \centering\SquareMp\ \rrSr\end{minipage} %} \hfill \newcommand{\textRubikSr}{\rrSr\ \rrhMp} \hfill 1991

20.21.58 Rotation Srp

These commands draw forms of the Singmaster Srp slice rotation.

\newcommand{\rrSrp}{\@rrrp{S}}\hfill 2000
\newcommand{\rrhSrp}{\rrhMp} 2001
20.21.59 Rotation Sf

\rrSf
\rrhSf
\RubikSf
\textRubikSf

These commands draw forms of the Singmaster Sf slice rotation. We need to just make square with Sf in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

20.21.60 Rotation Sfp

\rrSfp
\rrhSfp
\RubikSfp
\textRubikSfp

These commands draw forms of the Singmaster Sfp slice rotation. We need to just make square with Sfp in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

20.21.61 Rotation Sb

\rrSb
\rrhSb
\RubikSb
\textRubikSb

These commands draw forms of the Singmaster Sb slice rotation. We need to just make square with Sb in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.
\textbf{20.21.62 Rotation Sbp}

These commands draw forms of the Singmaster Sbp slice rotation. We need to just make square with Sbp in square; adjust box height using a \texttt{rule}; adjust \texttt{fboxsep} (default=3pt); adjust \texttt{fboxrule} (default=0.4pt); bounded by \{\} so no need to reset to defaults. Not visible from th front.

\begin{verbatim}
\newcommand{\rrSbp}{\@rrbp{S}}
\newcommand{\SquareSbp}{\@tlen\@SquareLetter{\rrSbp}\@tlen}
\newcommand{\rrhSbp}{\raisebox{-0.25mm}{\SquareSbp}}
\newcommand{\RubikSbp}{\raisebox{\@hRubik}{\SquareSbp}}
\newcommand{\textRubikSbp}{\rrhSbp}
\end{verbatim}

\textbf{20.21.63 Rotation U}

These commands all draw forms which denote the U rotation.

\begin{verbatim}
\newcommand{\rrU}{\@rr{U}}
\newcommand{\SquareU}{%\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw[thick, ->] (\@sb,\@sddd) -- (\@sbh, \@sddd);
\draw[thick] (\@sb,\@sdd) -- (\@sbh, \@sdd);
\draw[thick] (\@sb,\@sd) -- (\@sbh, \@sd);
\end{tikzpicture}%;}
\newcommand{\rrhU}{\raisebox{-0.333\height}{\@tlen\SquareU\@tlen}}
\newcommand{\RubikU}{%{\@rubikfont%\begin{minipage}{0.6cm}%\centering%\SquareU\ \rrU%\end{minipage}%;}
\newcommand{\textRubikU}{\rrU\ \rrhU}
\end{verbatim}

\textbf{20.21.64 Rotation Uw}

These commands all draw forms which denote the Uw rotation.

\begin{verbatim}
\newcommand{\rrUw}{\@rrw{U}}
\newcommand{\SquareUw}{%\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw[thick, ->] (\@sb,\@sddd) -- (\@sbh, \@sddd);
\draw[thick, ->] (\@sb,\@sdd) -- (\@sbh, \@sdd);
\draw[thick] (\@sb,\@sd) -- (\@sbh, \@sd);
\end{tikzpicture}%;}
\newcommand{\rrhUw}{\raisebox{-0.333\height}{\@tlen\SquareUw\@tlen}}
\newcommand{\RubikUw}{%{\@rubikfont%\begin{minipage}{0.6cm}%\centering%\SquareUw\ \rrUw%\end{minipage}%;}
\newcommand{\textRubikUw}{\rrUw\ \rrhUw}
\end{verbatim}
These commands all draw forms which denote the Up rotation.

\newcommand{\rrUp}{\@rrp{U}}
\newcommand{\SquareUp}{\begin{tikzpicture}\[scale=0.5\]DrawNotationBox;\draw [thick, ->] (@sb,0) -- (@sbh,0);\draw [thick] (@sb,0) -- (@sbh,0);\draw [thick] (@sb,0) -- (@sbh,0);\end{tikzpicture}}
\newcommand{\rrhUp}{\raisebox{-0.333\height}{\@tlen\SquareUp\@tlen}}
\newcommand{\RubikUp}{\@rubikfont\begin{minipage}{0.6cm}centering\SquareUp\rrUp\end{minipage}}
\newcommand{\textRubikUp}{\rrUp\,\rrhUp}

These commands all draw forms which denote the Uwp rotation.

\newcommand{\rrUwp}{\@rrwp{U}}
\newcommand{\SquareUwp}{\begin{tikzpicture}\[scale=0.5\]DrawNotationBox;\draw [thick, ->] (@sb,0) -- (@sbh,0);\draw [thick] (@sb,0) -- (@sbh,0);\draw [thick] (@sb,0) -- (@sbh,0);\end{tikzpicture}}
\newcommand{\rrhUwp}{\raisebox{-0.333\height}{\@tlen\SquareUwp\@tlen}}
\newcommand{\RubikUwp}{\@rubikfont\begin{minipage}{0.6cm}centering\SquareUwp\rrUp\end{minipage}}
\newcommand{\textRubikUwp}{\rrUp\,\rrhUwp}
These commands all draw forms which denote the Us rotation.

\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\@sb,\@sddd) -- (\@sbh, \@sddd);
\draw [thick] (\@sb,\@sdd) -- (\@sbh, \@sdd);
\draw [thick, <-] (\@sb,\@sd) -- (\@sbh, \@sd);
\end{tikzpicture}
\newcommand{\rrUs}{\textRubikUs}{\rrUs, \rrhUs}

These commands all draw forms which denote the Usp rotation.

\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <->] (\@sb,\@sddd) -- (\@sbh, \@sddd);
\draw [thick] (\@sb,\@sdd) -- (\@sbh, \@sdd);
\draw [thick, <-] (\@sb,\@sd) -- (\@sbh, \@sd);
\end{tikzpicture}
\newcommand{\rrUs}{\textRubikUs}{\rrUs, \rrhUs}
These commands all draw forms which denote the Ua rotation.

\newcommand{\rrUa}{\@rra{U}}
\newcommand{\SquareUa}{\begin{tikzpicture}[scale=0.5] \DrawNotationBox; \draw [thick, ->] (\@sb, \@sddd) -- (\@sbh, \@sddd); \draw [thick] (\@sb, \@sdd) -- (\@sbh, \@sdd); \draw [thick, ->] (\@sb, \@sd) -- (\@sbh, \@sd); \end{tikzpicture}}
\newcommand{\rrhUa}{\raisebox{-0.333\height}{\@tlen\SquareUa\@tlen}}
\newcommand{\RubikUa}{\@rubikfont\begin{minipage}{0.6cm} \centering \SquareUa\rrUa \end{minipage}}
\newcommand{\textRubikUa}{\rrUa\,\rrhUa}

These commands all draw forms which denote the Uap rotation.

\newcommand{\rrUap}{\@rrap{U}}
\newcommand{\SquareUap}{\begin{tikzpicture}[scale=0.5] \DrawNotationBox; \draw [thick, ->] (\@sb, \@sddd) -- (\@sbh, \@sddd); \draw [thick] (\@sb, \@sdd) -- (\@sbh, \@sdd); \draw [thick, ->] (\@sb, \@sd) -- (\@sbh, \@sd); \end{tikzpicture}}
\newcommand{\rrhUap}{\raisebox{-0.333\height}{\@tlen\SquareUap\@tlen}}
\newcommand{\RubikUap}{\@rubikfont\begin{minipage}{0.6cm} \centering \SquareUap\rrUap \end{minipage}}
\newcommand{\textRubikUap}{\rrUap\,\rrhUap}
\begin{tikzpicture}
\draw[thick,->] (\@sb,\@sddd) -- (\@sbh, \@sddd);
\draw [thick] (\@sb,\@sdd) -- (\@sbh, \@sdd);
\draw [thick,-<] (\@sb,\@sd) -- (\@sbh, \@sd);
\end{tikzpicture}

\newcommand{\rrhUap}{\raisebox{-0.333\height}{\@tlen\SquareUap\@tlen}}

\newcommand{\RubikUap}{\@rubikfont\begin{minipage}{0.6cm}\centering\SquareUap\rrUap\end{minipage}}
\newcommand{\textRubikUap}{\rrUap,\rrhUap}

\textbf{20.21.71 Rotations x and xp}
\begin{verbatim}
```
\rrx \rrhx \rubikx
\newcommand{\rrx}{\@rr{x}}
\newcommand{\Rubikx}{\@xyzRubik{x}}
\newcommand{\rrhx}{\@xyzh{x}}
```
\end{verbatim}

\textbf{20.21.72 Rotations y and yp}
\begin{verbatim}
```
\rry \rrhy \rubiky
\newcommand{\rry}{\@rr{y}}
\newcommand{\Rubiky}{\@xyzRubik{y}}
\newcommand{\rrhy}{\@xyzh{y}}
```
\end{verbatim}

\textbf{20.21.73 Rotations z and zp}
\begin{verbatim}
```
\rrz \rrhz \rubikz
\newcommand{\rrz}{\@rr{z}}
\newcommand{\rubikz}{\@xyzRubik{z}}
```
\end{verbatim}
These commands all draw forms which denote the zp rotation.

\rhzp
\Rubikzp
2216 \newcommand{\rrzp}{\@rrp{z}}
2217 \newcommand{\Rubikzp}{\@xyzRubikp{z}}
2218 \newcommand{\rrhzp}{\@xyzhp{z}}

20.21.74 Rotations u and d

\rhu
\Rubiku
2219 \newcommand{\rru}{\@rr{u}}
2220 \newcommand{\rrup}{\@rrp{u}}
2221 \newcommand{\rrhu}{\@xyzh{u}}
2222 \newcommand{\rrhup}{\@xyzhp{u}}
2223 \newcommand{\Rubiku}{\@xyzRubik{u}}
2224 \newcommand{\Rubikup}{\@xyzRubikp{u}}

\rhd
\Rubikd
2225 \newcommand{\rrd}{\@rr{d}}
2226 \newcommand{\rrdp}{\@rrp{d}}
2227 \newcommand{\rrhd}{\@xyzhbdfl{d}}
2228 \newcommand{\rrhdp}{\@xyzhbdflp{d}}
2229 \newcommand{\Rubikd}{\@xyzbdflRubik{d}}
2230 \newcommand{\Rubikdp}{\@xyzbdflRubikp{d}}

20.21.75 Rotations l and r

\rrl
\Rubikl
2231 \newcommand{\rrl}{\@rr{l}}
2232 \newcommand{\rrlp}{\@rrp{l}}
2233 \newcommand{\rrhl}{\@xyzhbdfl{l}}
2234 \newcommand{\rrhlp}{\@xyzhbdflp{l}}
2235 \newcommand{\Rubikl}{\@xyzbdflRubik{l}}
2236 \newcommand{\Rubiklp}{\@xyzbdflRubikp{l}}

\rrr
\Rubikr
2237 \newcommand{\rrr}{\@rr{r}}
2238 \newcommand{\rrrp}{\@rrp{r}}
2239 \newcommand{\rrhr}{\@xyzh{r}}
2240 \newcommand{\rrhrp}{\@xyzhp{r}}
2241 \newcommand{\Rubikr}{\@xyzRubik{r}}
2242 \newcommand{\Rubikrp}{\@xyzRubikp{r}}

20.21.76 Rotations f and b

\rrf
\Rubikf
2243 \newcommand{\rrf}{\@rr{f}}
2244 \newcommand{\rrfp}{\@rrp{f}}
2245 \newcommand{\rrhf}{\@xyzhbdfl{f}}
2246 \newcommand{\rrhfp}{\@xyzhbdflp{f}}
These commands all draw forms which denote the b and bp cube rotation.

\newcommand{\rrb}{\@rr{b}}
\newcommand{\rrbp}{\@rrp{b}}
\newcommand{\rrhb}{\@xyzhbdfl{b}}
\newcommand{\rrhbp}{\@xyzhbdflp{b}}
\newcommand{\Rubikb}{\@xyzbdflRubik{b}}
\newcommand{\Rubikbp}{\@xyzbdflRubikp{b}}

20.22 Face(c) and Face(m) rotations

This subsection (added Nov 2016) is to accommodate some additional notation, much used on the Jaap Puzzles website (Scherphuis J), which, although quite convenient, is technically ‘non-standard’. This additional notation makes available the (middle slice) rotations of the form Rm, Rmp, Lm, Lmp, ... and also the (whole cube) rotations Rc, Lc, ... (referenced to a face) as used on the Cube Lovers usenet group (1981–1997). This notation was probably invented by Singmaster (see Scherphuis J).

First we have some useful facilitating commands we shall make use of in conjunction with the Face(c) and Face(cp) notation.

\newcommand{\@xyzhc}[1]{\raisebox{-1.2pt}{\@rubikfont #1\@rubikfontFNS c}}
\newcommand{\@xyzhcp}[1]{\raisebox{-1.2pt}{\@rubikfont #1\@rubikprime\@rubikfontFNS c}}
\newcommand{\@xyzRubikc}[1]{\raisebox{3.45pt}{\@rubikfont #1\@rubikfontFNS c}}
\newcommand{\@xyzRubikcp}[1]{\raisebox{3.45pt}{\@rubikfont #1\@rubikprime\@rubikfontFNS c}}

20.22.1 Rotations Rc and Rcp

Whole cube rotations Rc = x, Rcp = xp.

These commands all draw forms which denote the Rc rotation.

\newcommand{\rrRc}{\@rrc{R}}
\newcommand{\RubikRc}{\@xyzRubikc{R}}
\newcommand{\rrhRc}{\@xyzhc{R}}

These commands all draw forms which denote the Rcp rotation.

\newcommand{\rrRcp}{\@rrcp{R}}
\newcommand{\RubikRcp}{\@xyzRubikcp{R}}
\newcommand{\rrhRcp}{\@xyzhcp{R}}
20.22.2 Rotations Lc and Lcp

Whole cube rotations Lc = xp, Lcp = x.

These commands all draw forms which denote the Lc rotation.

\rrLc
\rrhLc
\RubikLc

These commands all draw forms which denote the Lcp rotation.

\rrLcp
\rrhLcp
\RubikLcp

20.22.3 Rotations Uc and Ucp

Whole cube rotation Uc = y, Ucp = yp.

These commands all draw forms which denote the Uc rotation.

\rrUc
\rrhUc
\RubikUc

These commands all draw forms which denote the Ucp rotation.

\rrUcp
\rrhUcp
\RubikUcp

20.22.4 Rotations Dc and Dcp

Whole cube rotations Dc = yp, Dcp = y.

These commands all draw forms which denote the Dc rotation.

\rrDc
\rrhDc
\RubikDc

These commands all draw forms which denote the Dcp rotation.

\rrDcp
\rrhDcp
\RubikDcp

20.22.5 Rotations Fc and Fcp

Whole cube rotations Fc = z, Fcp = zp.

These commands all draw forms which denote the Fc rotation.

\rrFc
\rrhFc
\RubikFc

These commands all draw forms which denote the Fcp rotation.
\newcommand{\rrFcp}{\@rrcp{F}}
\newcommand{\RubikFcp}{\@xyzRubikcp{F}}
\newcommand{\rrhFcp}{\@xyzhcp{F}}

20.22.6 Rotations Bc and Bcp
Whole cube rotation Bc = zp, Bcp = z.

These commands all draw forms which denote the Bc rotation.
\newcommand{\rrBc}{\@rrc{B}}
\newcommand{\RubikBc}{\@xyzRubikc{B}}
\newcommand{\rrhBc}{\@xyzhc{B}}

These commands all draw forms which denote the Bcp rotation.
\newcommand{\rrBcp}{\@rrcp{B}}
\newcommand{\RubikBcp}{\@xyzRubikcp{B}}
\newcommand{\rrhBcp}{\@xyzhcp{B}}

20.22.7 Rotations Rm and Rmp
Rm = Mp = Sr (M follows Left). We use Sr and Srp as the templates.

These commands draw forms of the Rm slice rotation.
\newcommand{\rrRm}{\@rrm{R}}
\newcommand{\rrhRm}{\rrhMp}
\newcommand{\textRubikRm}{\@rubikfont
\begin{minipage}{0.6cm}
\centering
\SquareMp
\rrRm
\end{minipage}\
}
\newcommand{\textRubikRm}{\rrRm\rrhMp}

These commands draw forms of the Rmp slice rotation.
\newcommand{\rrRmp}{\@rrmp{R}}
\newcommand{\rrhRmp}{\rrhM}
\newcommand{\textRubikRmp}{\@rubikfont
\begin{minipage}{0.6cm}
\centering
\SquareM
\rrRmp
\end{minipage}\
}
\newcommand{\textRubikRmp}{\rrRmp\rrhM}
20.22.8 Rotations Lm and Lmp
Lm = M = Sl (M follows Left). We use Sl and Slp as the templates.

These commands draw forms of the Lm slice rotation.

\newcommand{\rrLm}{\@rrm{L}}
\newcommand{\rrhLm}{\rrhM}
\newcommand{\RubikLm}{\@rubikfont\SquareM\rrLm}
\newcommand{\textRubikLm}{\rrLm\rrhM}

These commands draw forms of the Lmp slice rotation.

\newcommand{\rrLmp}{\@rrmp{L}}
\newcommand{\rrhLmp}{\rrhMp}
\newcommand{\RubikLmp}{\@rubikfont\SquareMp\rrLmp}
\newcommand{\textRubikLmp}{\rrLmp\rrhMp}

20.22.9 Rotations Um and Ump
Um = Ep = Su (E follows Down). We use Su and Sup as the templates.

These commands draw forms of the Um slice rotation. We also need to fine-tune the spacing between these ‘slice’ hieroglyphs (especially Fs and Bs).

\newcommand{\rrUm}{\@rrm{U}}
\newcommand{\rrhUm}{\rrhEp}
\newcommand{\RubikUm}{\@rubikfont\SquareEp\rrUm}
\newcommand{\textRubikUm}{\rrUm\rrhEp}

These commands draw forms of the Ump slice rotation. We also need to fine-tune the spacing between these ‘slice’ hieroglyphs (especially Fs and Bs).

\newcommand{\rrUmp}{\@rrmp{U}}
\newcommand{\rrhUmp}{\rrhEp}
\newcommand{\RubikUmp}{\@rubikfont\SquareEp\rrUmp}
\newcommand{\textRubikUmp}{\rrUmp\rrhEp}
20.22.10 Rotations Dm and Dmp

Dm = E = Sd (E follows Down). We use Sd and Sdp as the templates.

These commands draw forms of the Singmaster Dm slice rotation.

\newcommand{\rrDm}{\@rrm{D}}
\newcommand{\rrhDm}{\rrhE}\
\newcommand{\RubikDm}{\@rubikfont\
\begin{minipage}{0.6cm}\
\centering\
\SquareE\\
\rrDm
\end{minipage}\
}}\
\newcommand{\textRubikDm}{\rrDm\,\rrhE}

20.22.11 Rotations Fm and Fmp

Fm = S = Sf (S follows Front). S is not visible from the front, so is represented in a square box. We use Sf and Sfp as the templates.

These commands draw forms of the Fm slice rotation. We need to just make square with Fm in square; adjust box height using a \rule; adjust \fboxsep

\newcommand{\rrFm}{\@rrmp{F}}
\newcommand{\rrhFm}{\rrhEp}\
\newcommand{\RubikFm}{\@rubikfont\
\begin{minipage}{0.6cm}\
\centering\
\SquareEp\\
\rrFm
\end{minipage}\
}}\
\newcommand{\textRubikFm}{\rrFm\,\rrhEp}
These commands draw forms of the Fmp slice rotation. We need to just make square with Fmp in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

These commands draw forms of the Bm slice rotation. We need to just make square with Bm in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

These commands draw forms of the Bmp slice rotation. We need to just make square with Bmp in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

This section relates to the recent Rubik notation developed by Randelshofer (see URL above), known as the “superset ENG rotations” (see Sections 9 and 9.9.1 for details).
I denote this notation as “CMST” (i.e., his C, M, S and T notation). Although this notation replicates previously existing World Cube Association notation, we include it here for completeness. Fortunately the Randelshofer notation does not conflict with existing notation.

Briefly, the CMST letters denote whole Cube, Middle, outer-Slice and wide (T) rotations associated with named faces. For example, the rotation-code MR indicates a rotation of the middle slice parallel to the RIGHT face, and in the same sense (direction) as \textbf{R} (see Section 9 for details).

First we make some useful facilitating commands for the CX1 and CX1p forms as these are handled with raised square brackets. We model these four macros on the equivalent \texttt{@xyzh}, \texttt{@xyzhp}, \texttt{@xyzRubik}, \texttt{@xyzRubikp} macros defined above.

\begin{verbatim}
\newcommand{\@randhc}[1]{[\raisebox{-1.2pt}{{\@rubikfont C#1}}]}
\newcommand{\@randhcp}[1]{[\raisebox{-1.2pt}{{\@rubikfont C#1\@rubikprime}}]}
\newcommand{\@randRubikc}[1]{\raisebox{3.45pt}{[\raisebox{-1.2pt}{{\@rubikfont C#1}}]}}
\newcommand{\@randRubikcp}[1]{\raisebox{3.45pt}{[\raisebox{-1.2pt}{{\@rubikfont C#1\@rubikprime}}]}}
\end{verbatim}

In the following we create the base form of a rotation code, eg [CR] and then the user typesets this rotation using CR as the argument for \texttt{rrh\{\}}. This works because the macro \texttt{@join{\{\}}} is used to join \texttt{rrh} and CR \rightarrow \texttt{rrhCR} etc.

The = signs at the beginning of each new section indicate the equivalence between the Randelshofer notation and the current (World Cube Association) notation.

### 20.23.1 Rotations CR and CRp

Whole cube rotations CR = Rc = x, CRp = Rcp = xp.

\begin{verbatim}
\rrCR \rrhCR \RubikCR
\rrCRp \rrhCRp \RubikCRp
\end{verbatim}

These commands all draw forms which denote the CR = Rc rotation.

These commands all draw forms which denote the CRp = Rcp rotation.

### 20.23.2 Rotations CL and CLp

Whole cube rotations CL = Lc = xp, CLp = Lcp = x.

\begin{verbatim}
\rrCL \rrhCL \RubikCL
\rrCLp \rrhCLp \RubikCLp
\end{verbatim}

These commands all draw forms which denote the CL rotation.
These commands all draw forms which denote the CLp rotation.

\newcommand{\rrCLp}{\@rrp{CL}}
\newcommand{\RubikCLp}{\@randRubikcp{L}}
\newcommand{\rrhCLp}{\@randhcp{L}}

20.23.3 Rotations CU and CUp

Whole cube rotation \( CU = Uc = y, CUp = Ucp = yp \).

These commands all draw forms which denote the CU rotation.

\newcommand{\rrCU}{\@rr{CU}}
\newcommand{\RubikCU}{\@randRubikc{U}}
\newcommand{\rrhCU}{\@randhc{U}}

20.23.4 Rotations CD and CDp

Whole cube rotations \( CD = Dc = yp, CDp = Dcp = y \).

These commands all draw forms which denote the CD rotation.

\newcommand{\rrCD}{\@rr{CD}}
\newcommand{\RubikCD}{\@randRubikc{D}}
\newcommand{\rrhCD}{\@randhc{D}}

20.23.5 Rotations CF and CFp

Whole cube rotations \( CF = Fc = z, CFp = Fcp = zp \).

These commands all draw forms which denote the CF rotation.

\newcommand{\rrCF}{\@rr{CF}}
\newcommand{\RubikCF}{\@randRubikc{F}}
\newcommand{\rrhCF}{\@randhc{F}}

These commands all draw forms which denote the CFp rotation.

\newcommand{\rrCFp}{\@rrp{CF}}
\newcommand{\RubikCFp}{\@randRubikcp{F}}
\newcommand{\rrhCFp}{\@randhcp{F}}
20.23.6 Rotations CB and CBp
Whole cube rotation \( \text{CB} = \text{Bc} = z_p, \text{CBp} = \text{Bcp} = z \).

These commands all draw forms which denote the CB rotation.

\begin{verbatim}
\newcommand{\rrCB}{\@rr{CB}}
\newcommand{\RubikCB}{\@randRubikc{B}}
\newcommand{\rrhCB}{\@randhc{B}}
\end{verbatim}

20.23.7 Rotations MR and MRp
\( \text{MR} = \text{Rm} = \text{M} = \text{Sr} \) (\( \text{M} \) follows Left). We use \( \text{Sr} \) and \( \text{SrR} \) as the templates.

These commands draw forms of the MR middle slice rotation.

\begin{verbatim}
\newcommand{\rrMR}{\@rr{MR}}
\newcommand{\rrhMR}{\rrhMp}
\newcommand{\RubikMR}{\@rubikfont\
\begin{minipage}{0.6cm}
\centering
\SquareMp\\
\rrMR\end{minipage}}
\newcommand{\textRubikMR}{\rrMR\rrhMP}
\end{verbatim}

20.23.8 Rotations ML and MLp
\( \text{ML} = \text{Lm} = \text{M} = \text{Sl} \) (\( \text{M} \) follows Left). We use \( \text{Sl} \) and \( \text{Slp} \) as the templates.

These commands draw forms of the MRp slice rotation.

\begin{verbatim}
\newcommand{\rrMRp}{\@rrp{MR}}
\newcommand{\rrhMRp}{\rrhM}
\newcommand{\RubikMRp}{\@rubikfont\
\begin{minipage}{0.6cm}
\centering
\SquareM\\
\rrMRp\end{minipage}}
\newcommand{\textRubikMRp}{\rrMRp\rrhM}
\end{verbatim}
These commands draw forms of the ML slice rotation.
\newcommand{\rrML}{\@rr{ML}}
\newcommand{\rrhML}{\rrhM}
\newcommand{\RubikML}{\@rubikfont
\begin{minipage}{0.6cm}
\centering
\SquareM\%
\end{minipage}%%
\end{minipage}}
\newcommand{\textRubikML}{\rrML\,\rrhM}

These commands draw forms of the MLp slice rotation.
\newcommand{\rrMLp}{\@rrp{ML}}
\newcommand{\rrhMLp}{\rrhMp}
\newcommand{\RubikMLp}{\@rubikfont
\begin{minipage}{0.6cm}
\centering
\SquareMp\%
\end{minipage}%%
\end{minipage}}
\newcommand{\textRubikMLp}{\rrMLp\,\rrhMp}

20.23.9 Rotations MU and MUp
MU = Um = Ep = Su (E follows Down). We use Su and Sup as the templates.

These commands draw forms of the MU slice rotation.
\newcommand{\rrMU}{\@rr{MU}}
\newcommand{\rrhMU}{\rrhEp}
\newcommand{\RubikMU}{\@rubikfont
\begin{minipage}{0.6cm}
\centering
\SquareEp\%
\end{minipage}%%
\end{minipage}}
\newcommand{\textRubikMU}{\rrMU\,\rrhEp}

These commands draw forms of the MUp slice rotation.
\newcommand{\rrMUp}{\@rrp{MU}}
\newcommand{\rrhMUp}{\rrhE}
\newcommand{\RubikMUp}{\@rubikfont
\begin{minipage}{0.6cm}
\centering
\SquareE\%
\end{minipage}%%
\end{minipage}}
\newcommand{\textRubikMUp}{\rrMUp\,\rrhE}
20.23.10 Rotations MD and MDp

MD = Dm = E = Sd (E follows Down). We use Sd and Sdp as the templates.

These commands draw forms of the Randelshofer MD slice rotation.

\newcommand{\textRubikMD}{\textRubikMUp}
\newcommand{\textRubikMDp}{\textRubikMDp}

20.23.11 Rotations MF and MFp

MF = Fm = S = Sf (S follows Front). MF = S is not visible from the front, so is represented in a square box. We use Sf and Sfp as the templates.

These commands draw forms of the MF slice rotation. We need to just make square with MF in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

\newcommand{\textRubikMF}{\textRubikMFp}
rubikcube  (Rubik bundle v5.0, 2018)  www.ctan.org/pkg/rubik

\newcommand{\RubikMF}{\raisebox{-0.25mm}{\SquareMFp}}
\newcommand{\textRubikMFp}{\rrhMFp}
\newcommand{\RubikMFp}{\raisebox{\@hRubik}{\SquareMFp}}
\newcommand{\textRubikMFp}{\rrhMFp}

These commands draw forms of the MFp slice rotation. We need to just make square with MFp in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

\newcommand{\rrMB}{\@rr{MB}}
\newcommand{\SquareMB}{\@tlen\@SquareLetter{\rrMB}\@tlen}
\newcommand{\rrhMB}{\raisebox{-0.25mm}{\SquareMB}}
\newcommand{\RubikMB}{\raisebox{\@hRubik}{\SquareMB}}
\newcommand{\textRubikMB}{\rrhMB}

These commands draw forms of the MB slice rotation. We need to just make square with MB in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

\newcommand{\rrMBp}{\@rrp{MB}}
\newcommand{\SquareMBp}{\@tlen\@SquareLetter{\rrMBp}\@tlen}
\newcommand{\rrhMBp}{\raisebox{-0.25mm}{\SquareMBp}}
\newcommand{\RubikMBp}{\raisebox{\@hRubik}{\SquareMBp}}
\newcommand{\textRubikMBp}{\rrhMBp}

These commands draw forms of the MBp slice rotation. We need to just make square with MBp in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

\newcommand{\@rrT}{\@rubikfont T} %
\newcommand{\@rrhT}{\@join{\rrh}\{w\}} %
\newcommand{\@textRubikT}{\@join{\textRubik}\{w\}} %

20.23.12  Rotations MB and MBp

MB = Bm = Sp = Sb. We use Sb and Sbp as the templates.

\newcommand{\rrpT}{\@rrp{T}} %
\newcommand{\textRubikT}{\@join{\textRubik}\{w\}} %

20.23.13  Rotations T (wide)

TR = Rw  (We use Rw and Rwp as the templates).
We first set up some useful T and Tp macros for use with (L,R,U,D,F,B) slightly more general than before. As before the rubikfont command needs to be delimited using two curly brackets to contain the font expression.
These commands draw forms of the TL = Lw wide slice rotation.
\newcommand{\rrTL}{\@rrT{L}} % = rrLw
\newcommand{\rrhTL}{\@rrhT{L}}
\newcommand{\textRubikTL}{\@textRubikT{L}}
\newcommand{\RubikTL}{\@RubikT{L}}

These commands draw forms of the TLP = Lwp wide slice rotation.
\newcommand{\rrTP}{\@rrTp{L}}
\newcommand{\rrhTP}{\@rrhTp{L}}
\newcommand{\textRubikTP}{\@textRubikTp{L}}
\newcommand{\RubikTP}{\@RubikTp{L}}

These commands draw forms of the TR = Rw wide slice rotation.
\newcommand{\rrTR}{\@rrT{R}}
\newcommand{\rrhTR}{\@rrhT{R}}
\newcommand{\textRubikTR}{\@textRubikT{R}}
\newcommand{\RubikTR}{\@RubikT{R}}

These commands draw forms of the TRP = Rmp wide slice rotation.
\newcommand{\rrTRP}{\@rrTp{R}}
\newcommand{\rrhTRP}{\@rrhTp{R}}
\newcommand{\textRubikTRP}{\@textRubikTp{R}}
\newcommand{\RubikTRP}{\@RubikTp{R}}

These commands draw forms of the TU = Uw wide slice rotation.
\newcommand{\rrTU}{\@rrT{U}}
\newcommand{\rrhTU}{\@rrhT{U}}
\newcommand{\textRubikTU}{\@textRubikT{U}}
\newcommand{\RubikTU}{\@RubikT{U}}
\rrTUp \textRubikTUp \RubikTUp
These commands draw forms of the \TUp = Uwp wide slice rotation.

\rrhTUp \textRubikTPU \RubikTPU
\newcommand{\rrhTUp}{\@rrhTp{U}}
\newcommand{\textRubikTUp}{\@textRubikTp{U}}
\newcommand{\RubikTUp}{\@RubikTp{U}}

\rrTD \textRubikTD \RubikTD
These commands draw forms of the \TD = Dw wide slice rotation.
\newcommand{\rrTD}{\@rrT{D}}
\newcommand{\textRubikTD}{\@textRubikT{D}}
\newcommand{\RubikTD}{\@RubikT{D}}

\rrTDp \textRubikTDp \RubikTDp
These commands draw forms of the \TDp = Dwp wide slice rotation.
\newcommand{\rrTDp}{\@rrTp{D}}
\newcommand{\textRubikTDp}{\@textRubikTp{D}}
\newcommand{\RubikTDp}{\@RubikTp{D}}

\rrTF \textRubikTF \RubikTF
These commands draw forms of the \TF = Fw wide slice rotation.
\newcommand{\rrTF}{\@rrT{F}}
\newcommand{\textRubikTF}{\@textRubikT{F}}
\newcommand{\RubikTF}{\@RubikT{F}}

\rrTFp \textRubikTFp \RubikTFp
These commands draw forms of the \TFp = Fwp wide slice rotation.
\newcommand{\rrTFp}{\@rrTp{F}}
\newcommand{\textRubikTFp}{\@textRubikTp{F}}
\newcommand{\RubikTFp}{\@RubikTp{F}}

\rrTB \textRubikTB \RubikTB
These commands draw forms of the \TB = Bw wide slice rotation. NOTE we just
use the letters TB in a square
\newcommand{\rrTB}{\@rrT{B}}
\newcommand{\SquareTB}{\@tlen\@SquareLetter{\rrTB}\@tlen}
\newcommand{\rrhTB}{\raisebox{-0.25mm}{\SquareTB}}
\newcommand{\textRubikTB}{\rrhTB}
\newcommand{\RubikTB}{\raisebox{\@hRubik}{\SquareTB}}

\rrTBp \textRubikTBp \RubikTBp
These commands draw forms of the \TBp = Bwp wide slice rotation. NOTE we
just use the letters ‘TB’ in a square
\newcommand{\rrTBp}{\@rrTp{B}}
\newcommand{\SquareTBp}{\@tlen\@SquareLetter{\rrTBp}\@tlen}
\newcommand{\rrhTBp}{\raisebox{-0.25mm}{\SquareTBp}}
\newcommand{\textRubikTBp}{\rrhTBp}
\newcommand{\RubikTBp}{\raisebox{\@hRubik}{\SquareTBp}}
20.23.14 Rotations SR and SRp (opposite slices)

opposite slices in same direction \( SR = Rs \) (both rotating in R direction)

\begin{verbatim}
\rrSR\newcommand{\rrSR}{\@rr{SR}}\newcommand{\rrhSR}{\rrhRs}\newcommand{\RubikSR}{\@rubikfont\begin{minipage}{0.6cm}\centering\SquareRs\end{minipage}}\newcommand{\textRubikSR}{\rrSR\,\rrhRs}
\rrSRp\newcommand{\rrSRp}{\@rrp{SR}}\newcommand{\rrhSRp}{\rrhRsp}\newcommand{\RubikSRp}{\@rubikfont\begin{minipage}{0.6cm}\centering\SquareRsp\end{minipage}}\newcommand{\textRubikSRp}{\rrSRp\,\rrhRsp}
\end{verbatim}

These commands draw forms of the SR opposite slice rotation.

20.23.15 Rotations SL and SLp

SL = Ls (both rotating in L direction)

\begin{verbatim}
\rrSL\newcommand{\rrSL}{\@rr{SL}}\newcommand{\rrhSL}{\rrhLs}\newcommand{\RubikSL}{\@rubikfont\begin{minipage}{0.6cm}\centering\SquareLs\end{minipage}}\newcommand{\textRubikSL}{\rrSL\,\rrhLs}
\rrSLp\newcommand{\rrSLp}{\@rrp{SL}}\newcommand{\rrhSLp}{\rrhLsp}\newcommand{\RubikSLp}{\@rubikfont\begin{minipage}{0.6cm}\centering\SquareLsp\end{minipage}}\newcommand{\textRubikSLp}{\rrSLp\,\rrhLsp}
\end{verbatim}

These commands draw forms of the SL opposite slice rotation.
20.23.16 Rotations SU and SUp

SU = Us (both rotating in U direction)

These commands draw forms of the SU opposite slice rotation.

\[ \text{SU} \]
\[ \text{SUp} \]

20.23.17 Rotations SD and SDp

SD = Ds (both rotating in D direction)

These commands draw forms of the SD opposite slice rotation.
These commands draw forms of the SDp opposite slice rotation.

\newcommand{\textRubikSDp}{\rrSDp\,\rrhSDp}

These commands draw forms of the SF opposite slice rotation. We need to just make square with SF in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

\newcommand{\textRubikSF}{\rrhSF}

These commands draw forms of the SFp opposite slice rotation. We need to just make square with SFp in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

\newcommand{\textRubikSFp}{\rrhSFp}

20.23.18 Rotations SF and SFp

SF = Fs is not visible from the front, so is represented in a square box. Both rotating in the F direction. We use Fs and Fsp as the templates.

\newcommand{\textRubikSFp}{\rrhSFp}
20.23.19 Rotations SB and SBp

SB = Bs is not visible from the front, so is represented in a square box. Both rotating in the B direction We use Bs and Bsp as the templates.

These commands draw forms of the SB opposite slice rotation. We need to just make square with SB in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

\rrSB \rrhSB \RubikSB \textRubikSB

These commands draw forms of the SBp opposite slice rotation. We need to just make square with SBp in square; adjust box height using a \rule; adjust \fboxsep (default=3pt); adjust \fboxrule (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

\rrSBp \rrhSBp \RubikSBp \textRubikSBp

20.24 Axis rotations (textRubik versions)

For completeness we include a \textRubik version of all the axis rotation codes (making them equal to their hieroglyphic \rrh version). Obviously this list must go at the end of this file. While these commands are perhaps strictly unnecessary, the motivation is to allow users to include them in a \ShowSequence command when using the \textRubik font argument.

\textRubikx \textRubikxp \textRubiky \textRubikyp \textRubikz \textRubikzp \textRubikl \textRubiklp \textRubikr \textRubikrp \textRubiku \textRubikuup \textRubikd \textRubikdp \textRubikf \textRubikfp
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.

Symbols

<table>
<thead>
<tr>
<th>Symbols</th>
<th>1701, 1722, 2158, 2179,</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>1743, 1765, 2200, 2309,</td>
</tr>
<tr>
<td>,</td>
<td>1786, 1807, 2320, 2331,</td>
</tr>
<tr>
<td>\</td>
<td>1828, 1849, 2342, 2353,</td>
</tr>
<tr>
<td>1244, 1265, 1286, 1307, 1349, 1391, 1433, 1471, 1512, 1575, 1617, 1659,</td>
<td>1412, 1452, 1492, 1554, 1596, 1638, 1680,</td>
</tr>
<tr>
<td>73</td>
<td>2158, 2200, 2320, 2342,</td>
</tr>
<tr>
<td>2106, 2127</td>
<td>1429, 1448, 1238, 1248</td>
</tr>
<tr>
<td>2148, 2169</td>
<td>1467, 1488, 1259, 1269</td>
</tr>
<tr>
<td>2190, 2388</td>
<td>1508, 1550, 1280, 1290</td>
</tr>
<tr>
<td>2393, 2398</td>
<td>1571, 1592, 1301, 1311</td>
</tr>
<tr>
<td>2403, 2538</td>
<td>1613, 1634, 1322, 1332</td>
</tr>
<tr>
<td>2543, 2548</td>
<td>1655, 1676, 1343, 1353</td>
</tr>
<tr>
<td>2553, 2623</td>
<td>1697, 1718, 1364, 1374</td>
</tr>
<tr>
<td>2628, 2721</td>
<td>1739, 1761, 1385, 1395</td>
</tr>
<tr>
<td>2726, 2731, 2736</td>
<td>1782, 1803, 1406, 1416</td>
</tr>
</tbody>
</table>

\@xyzRubik ...
2106, 2127, 1429, 1448, 1238, 1248
2202, 2208, 1467, 1488, 1259, 1269
2214, 2223, 2241, 1508, 1550, 1280, 1290
2270, 2276, 1571, 1592, 1301, 1311
2282, 2288, 2294, 1613, 1634, 1322, 1332
2287, 2291, 2297, 1655, 1676, 1343, 1353
2290, 2298, 2305, 1697, 1718, 1364, 1374
2302, 2308, 2311, 1739, 1761, 1385, 1395
2316, 2327, 1782, 1803, 1406, 1416
2324, 2328, 1824, 1845, 1427, 1437
2329, 2331, 1866, 1887, 1446, 1456
2332, 2333, 1908, 1929, 1465, 1475
2336, 2337, 1940, 1951, 1486, 1496
2346, 2347, 1962, 1973, 1506, 1517
2348, 2349, 1984, 1995, 1528, 1538
2352, 2356, 2006, 2047, 1549, 1579
2353, 2354, 2069, 2091, 1590, 1600
2358, 2359, 2112, 2133, 1611, 1621
2360, 2361, 2154, 2175, 1632, 1642
2362, 2363, 2196, 2305, 1653, 1663
2364, 2365, 2316, 2327, 1674, 1684
2366, 2367, 2338, 2349, 1695, 1705
2368, 2369, 2359, 2371, 1716, 1726
2370, 2371, 2382, 2455, 1737, 1747
2372, 2373, 2466, 2477, 1759, 1769
2374, 2375, 2488, 2499, 1780, 1790
2376, 2377, 2510, 2521, 1801, 1811
2378, 2379, 2532, 2565, 1822, 1832
2380, 2381, 2578, 2638, 1843, 1853
2382, 2383, 2649, 2660, 1864, 1874
2384, 2385, 2671, 2682, 1885, 1895
2386, 2387, 2693, 2704, 1906, 1927
2388, 2389, 2715, 1938, 1949
2390, 2391, 1960, 1971
2392, 2393, 1982, 1993
2394, 2395, 2004, 2034
2396, 2397, 2045, 2055
2398, 2399, 2067, 2078
2400, 2401, 2089, 2099
2402, 2403, 2110, 2120
2404, 2405, 2131, 2141
2406, 2407, 2152, 2162
2408, 2409, 2173, 2183
2410, 2411, 2194, 2303
2412, 2413, 2314, 2325
2414, 2415, 2336, 2347
2416, 2417, 34, 45, 47, 1086
2418, 2419, 1240, 1261, 1282, 1088, 1095
2420, 2421, 1303, 1324, 1097, 1104
2422, 2423, 1345, 1366, 1106, 1113
2424, 2425, 1387, 1408, 1115, 1227
2426, 2427, 2486, 2497
Index

480, 483, 484, 486, 487, 489, 490
\ddy 460, 463, 464, 466, 467, 479, 480, 483, 484, 486, 487, 489, 490
\definecolor 17, 18, 19, 20
\Dlb 96, 131, 321, 368, 482, 715, 736
\Dlm 95, 130, 314, 361, 472, 716
\Dlt 94, 129, 307, 354, 462, 677, 717, 869
\Dm 26
\Dmb 96, 131, 323, 370, 485, 735
\Dmm 95, 130, 316, 363, 475
\Dmt 94, 129, 309, 356, 465, 678, 870
\Dp 25
\draw 212, 214, 216, 219, 221, 223, 226, 228, 230, 237, 239, 241, 244, 246, 248, 251, 253, 255, 259, 261, 263, 266, 268, 270, 273, 275, 277, 285
\ddy 460, 463, 464, 466, 467, 479, 480, 483, 484, 486, 487, 489, 490
\definecolor 17, 18, 19, 20
\Dlb 96, 131, 321, 368, 482, 715, 736
\Dlm 95, 130, 314, 361, 472, 716
\Dlt 94, 129, 307, 354, 462, 677, 717, 869
\Dm 26
\Dmb 96, 131, 323, 370, 485, 735
\Dmm 95, 130, 316, 363, 475
\Dmt 94, 129, 309, 356, 465, 678, 870
\Dp 25
\draw 212, 214, 216, 219, 221, 223, 226, 228, 230, 237, 239, 241, 244, 246, 248, 251, 253, 255, 259, 261, 263, 266, 268, 270, 273, 275, 277, 285
\ddy 460, 463, 464, 466, 467, 479, 480, 483, 484, 486, 487, 489, 490
\definecolor 17, 18, 19, 20
\Dlb 96, 131, 321, 368, 482, 715, 736
\Dlm 95, 130, 314, 361, 472, 716
\Dlt 94, 129, 307, 354, 462, 677, 717, 869
\Dm 26
\Dmb 96, 131, 323, 370, 485, 735
\Dmm 95, 130, 316, 363, 475
\Dmt 94, 129, 309, 356, 465, 678, 870
\Dp 25
\draw 212, 214, 216, 219, 221, 223, 226, 228, 230, 237, 239, 241, 244, 246, 248, 251, 253, 255, 259, 261, 263, 266, 268, 270, 273, 275, 277, 285
\ddy 460, 463, 464, 466, 467, 479, 480, 483, 484, 486, 487, 489, 490
\definecolor 17, 18, 19, 20
\Dlb 96, 131, 321, 368, 482, 715, 736
\Dlm 95, 130, 314, 361, 472, 716
\Dlt 94, 129, 307, 354, 462, 677, 717, 869
\Dm 26
\Dmb 96, 131, 323, 370, 485, 735
\Dmm 95, 130, 316, 363, 475
\Dmt 94, 129, 309, 356, 465, 678, 870
\Dp 25
\draw 212, 214, 216, 219, 221, 223, 226, 228, 230, 237, 239, 241, 244, 246, 248, 251, 253, 255, 259, 261, 263, 266, 268, 270, 273, 275, 277, 285
\DrawRubikFaceBack . \DrawRubikFlatUp ... 1305, 1316.
.... 41, 762, 768  .... 44, 423, 433.
\DrawRubikFaceBackSide 423, 630, 644, 757 .... 1347, 1358.
.... 41, 719, 774 \Drb ... 96, 131, 325, 1368, 1379.
\DrawRubikFaceBS .. 774 372, 488, 698, 734 .... 1389, 1400.
\DrawRubikFaceD ... 764 \Drm ....... 95, 130, 1410, 1421.
\DrawRubikFaceDown ... 41, 758, 764 318, 365, 478, 697 
.... 41, 738, 770 \Drt 94, 129, 311, 358, 1431, 1440.
\DrawRubikFaceDownSide 468, 679, 696, 871 .... 1450, 1459.
.... 41, 738, 770 \Ds ............ 27 1469, 1480.
\DrawRubikFaceDS .. 770 \Dw ............ 27 .... 1490, 1500.
\DrawRubikFaceF ... 767 \dx 790, 792, 793, 799, 1552, 1563.
\DrawRubikFaceFront 801, 802, 808, 1573, 1584.
.... 41, 761, 767 810, 811, 817, 1594, 1605.
\DrawRubikFaceFrontSide 819, 820, 828, 1615, 1626.
.... 41, 662, 773 830, 831, 847, 1636, 1647.
\DrawRubikFaceFS ... 773 849, 850, 884, 1657, 1668.
\DrawRubikFaceL ... 765 886, 887, 903, 1678, 1689.
\DrawRubikFaceLeft ... 41, 759, 765 905, 906, 938, 1699, 1710.
\DrawRubikFaceLeftSide ... 41, 700, 771 940, 941, 955, 1720, 1731.
\DrawRubikFaceLS ... 771 957, 958, 978, 1741, 1753.
\DrawRubikFaceR ... 766 980, 981, 995, 1763, 1774.
\DrawRubikFaceRight \dy 997, 998, 1016, 1784, 1795.
.... 41, 760, 766 1027, 1032, 1035 1805, 1816.
\DrawRubikFaceRightSide \dx 791, 792, 793, 800, 1826, 1837.
.... 41, 681, 772 801, 802, 809, 1847, 1858.
\DrawRubikFaceRS .. 772 810, 811, 818, 1868, 1879.
\DrawRubikFaceU ... 763 819, 820, 829, 1889, 1900.
\DrawRubikFaceUp ... 41, 757, 763 830, 831, 848, 1910, 1931.
\DrawRubikFaceUpSide ... 41, 643, 769 849, 850, 885, 1942, 1953.
... 44, 423, 458, 423, 595, 423, 595, 635, 641, 720, 762 
.... 44, 423, 458, 631, 639, 739, 758 \E ............ 26 2107, 2118.
\DrawRubikFlatDown . 1022, 1027, 1032, 1035 2156, 2167.
.... 44, 423, 458, 44, 423, 458, 423, 595, 635, 641, 720, 762 
.... 44, 423, 458, 631, 639, 739, 758 \E ............ 26 2198, 2209.
\DrawRubikFlatFront \end ............ 36, 2219, 2307.
.... 44, 423, 563, 633, 663, 761 37, 49, 50, 1091, 2318, 2329.
\DrawRubikFlatLeft . 1092, 1100, 2340, 2351.
.... 44, 423, 493, 1101, 1109, 2362, 2373.
632, 640, 701, 759 1110, 1118, 2384, 2457.
\DrawRubikFlatRight ... 44, 423, 1119, 1232, 2468, 2479.
528, 634, 682, 760 1242, 1253, 2490, 2501.
\DrawRubikFlatRight ... 44, 423, 1263, 1274, 2512, 2523.
528, 634, 682, 760 1284, 1295, 2534, 2567.
\DrawRubikFlatRight ... 44, 423, 528, 634, 682, 760 2580, 2640.
<table>
<thead>
<tr>
<th>Index</th>
<th>H</th>
<th>Lrb</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>\height</td>
<td>\height</td>
</tr>
<tr>
<td></td>
<td>1234,</td>
<td>207,</td>
</tr>
<tr>
<td></td>
<td>1255,</td>
<td>350,</td>
</tr>
<tr>
<td></td>
<td>1267,</td>
<td>397,</td>
</tr>
<tr>
<td></td>
<td>1297,</td>
<td>523,</td>
</tr>
<tr>
<td></td>
<td>1318,</td>
<td>669,</td>
</tr>
<tr>
<td></td>
<td>1339,</td>
<td>747,</td>
</tr>
<tr>
<td></td>
<td>1360,</td>
<td>863,</td>
</tr>
<tr>
<td></td>
<td>1381,</td>
<td>191,</td>
</tr>
<tr>
<td></td>
<td>1402,</td>
<td>199,</td>
</tr>
<tr>
<td></td>
<td>1423,</td>
<td>343,</td>
</tr>
<tr>
<td></td>
<td>1442,</td>
<td>390,</td>
</tr>
<tr>
<td></td>
<td>1461,</td>
<td>513,</td>
</tr>
<tr>
<td></td>
<td>1482,</td>
<td>670,</td>
</tr>
<tr>
<td></td>
<td>1502,</td>
<td>862,</td>
</tr>
<tr>
<td></td>
<td>1544,</td>
<td>183,</td>
</tr>
<tr>
<td></td>
<td>1586,</td>
<td>336,</td>
</tr>
<tr>
<td></td>
<td>1607,</td>
<td>383,</td>
</tr>
<tr>
<td></td>
<td>1628,</td>
<td>503,</td>
</tr>
<tr>
<td></td>
<td>1649,</td>
<td>650,</td>
</tr>
<tr>
<td></td>
<td>1670,</td>
<td>671,</td>
</tr>
<tr>
<td></td>
<td>1691,</td>
<td>861,</td>
</tr>
<tr>
<td></td>
<td>1712,</td>
<td>27,</td>
</tr>
<tr>
<td></td>
<td>1733,</td>
<td>\lx</td>
</tr>
<tr>
<td></td>
<td>1755,</td>
<td>\lx</td>
</tr>
<tr>
<td></td>
<td>1776,</td>
<td>\lx</td>
</tr>
<tr>
<td></td>
<td>1797,</td>
<td>\lx</td>
</tr>
<tr>
<td></td>
<td>1818,</td>
<td>\lx</td>
</tr>
<tr>
<td></td>
<td>1839,</td>
<td>494,</td>
</tr>
<tr>
<td></td>
<td>1860,</td>
<td>498,</td>
</tr>
<tr>
<td></td>
<td>1881,</td>
<td>499,</td>
</tr>
<tr>
<td></td>
<td>1902,</td>
<td>501,</td>
</tr>
<tr>
<td></td>
<td>2041,</td>
<td>502,</td>
</tr>
<tr>
<td></td>
<td>2063,</td>
<td>504,</td>
</tr>
<tr>
<td></td>
<td>2085,</td>
<td>505,</td>
</tr>
<tr>
<td></td>
<td>2106,</td>
<td>508,</td>
</tr>
<tr>
<td></td>
<td>2127,</td>
<td>509,</td>
</tr>
<tr>
<td></td>
<td>2148,</td>
<td>511,</td>
</tr>
<tr>
<td></td>
<td>2169,</td>
<td>512,</td>
</tr>
<tr>
<td></td>
<td>2190</td>
<td>514,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>515,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>518,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>519,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>521,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>522,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>524,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>525,</td>
</tr>
<tr>
<td>I</td>
<td>\ifthenelse</td>
<td>56,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>62,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>67,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>788,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>797,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>806,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>815,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>826,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>845,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>882,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>901,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>936,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>953,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>976,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>993,</td>
</tr>
<tr>
<td>L</td>
<td>\La</td>
<td>29,</td>
</tr>
<tr>
<td></td>
<td>\Lc</td>
<td>29,</td>
</tr>
<tr>
<td></td>
<td>\Ld</td>
<td>28,</td>
</tr>
<tr>
<td></td>
<td>\Lh</td>
<td>27,</td>
</tr>
<tr>
<td></td>
<td>\Ll</td>
<td>27,</td>
</tr>
<tr>
<td></td>
<td>\Lr</td>
<td>25,</td>
</tr>
<tr>
<td></td>
<td>\Lw</td>
<td>27,</td>
</tr>
<tr>
<td></td>
<td>\Lx</td>
<td>25,</td>
</tr>
<tr>
<td></td>
<td>\Ly</td>
<td>26,</td>
</tr>
<tr>
<td></td>
<td>\Lz</td>
<td>26,</td>
</tr>
<tr>
<td></td>
<td>\Lb</td>
<td>26,</td>
</tr>
<tr>
<td></td>
<td>\Lm</td>
<td>26,</td>
</tr>
<tr>
<td></td>
<td>\Lp</td>
<td>26,</td>
</tr>
<tr>
<td></td>
<td>\Lq</td>
<td>26,</td>
</tr>
<tr>
<td></td>
<td>\Lt</td>
<td>26,</td>
</tr>
<tr>
<td>N</td>
<td>\ncubes</td>
<td>1007,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1010,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1011,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1012,</td>
</tr>
<tr>
<td></td>
<td>\NeedsTeXFormat</td>
<td>4,</td>
</tr>
<tr>
<td></td>
<td>\newcommand</td>
<td>15,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>24,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>26,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>26,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>26,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>26,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>26,</td>
</tr>
<tr>
<td>173, 174, 175</td>
<td>1142, 1143</td>
<td>1349, 1350</td>
</tr>
<tr>
<td>176, 177, 178</td>
<td>1144, 1145</td>
<td>1352, 1360</td>
</tr>
<tr>
<td>182, 186, 190</td>
<td>1146, 1147</td>
<td>1362, 1370</td>
</tr>
<tr>
<td>194, 198, 202</td>
<td>1148, 1149</td>
<td>1371, 1373</td>
</tr>
<tr>
<td>206, 210, 233</td>
<td>1150, 1151</td>
<td>1381, 1383</td>
</tr>
<tr>
<td>280, 281, 328</td>
<td>1152, 1153</td>
<td>1391, 1392</td>
</tr>
<tr>
<td>375, 423, 458</td>
<td>1154, 1155</td>
<td>1394, 1402</td>
</tr>
<tr>
<td>493, 528, 563</td>
<td>1156, 1157</td>
<td>1404, 1412</td>
</tr>
<tr>
<td>595, 629, 637</td>
<td>1158, 1159</td>
<td>1413, 1415</td>
</tr>
<tr>
<td>643, 662, 681</td>
<td>1160, 1162</td>
<td>1423, 1425</td>
</tr>
<tr>
<td>700, 719, 738</td>
<td>1164, 1166</td>
<td>1433, 1434</td>
</tr>
<tr>
<td>757, 758, 759</td>
<td>1168, 1170</td>
<td>1436, 1442</td>
</tr>
<tr>
<td>760, 761, 762</td>
<td>1172, 1177</td>
<td>1444, 1452</td>
</tr>
<tr>
<td>763, 764, 765</td>
<td>1178, 1180</td>
<td>1453, 1455</td>
</tr>
<tr>
<td>766, 767, 768</td>
<td>1181, 1182</td>
<td>1461, 1463</td>
</tr>
<tr>
<td>769, 770, 771</td>
<td>1183, 1184</td>
<td>1471, 1472</td>
</tr>
<tr>
<td>772, 773, 774</td>
<td>1185, 1186</td>
<td>1474, 1482</td>
</tr>
<tr>
<td>775, 776, 777</td>
<td>1187, 1188</td>
<td>1484, 1492</td>
</tr>
<tr>
<td>781, 782, 783</td>
<td>1189, 1190</td>
<td>1493, 1495</td>
</tr>
<tr>
<td>785, 786, 795</td>
<td>1191, 1192</td>
<td>1502, 1504</td>
</tr>
<tr>
<td>804, 813, 822</td>
<td>1193, 1194</td>
<td>1512, 1514</td>
</tr>
<tr>
<td>833, 838, 839</td>
<td>1195, 1196</td>
<td>1515, 1516</td>
</tr>
<tr>
<td>840, 841, 852</td>
<td>1197, 1198</td>
<td>1517, 1518</td>
</tr>
<tr>
<td>857, 858, 859</td>
<td>1199, 1200</td>
<td>1519, 1520</td>
</tr>
<tr>
<td>860, 865, 866</td>
<td>1201, 1202</td>
<td>1521, 1522</td>
</tr>
<tr>
<td>867, 868, 873</td>
<td>1203, 1204</td>
<td>1523, 1524</td>
</tr>
<tr>
<td>874, 875, 876</td>
<td>1205, 1206</td>
<td>1525, 1526</td>
</tr>
<tr>
<td>877, 878, 889</td>
<td>1207, 1208</td>
<td>1527, 1528</td>
</tr>
<tr>
<td>894, 895, 896</td>
<td>1209, 1210</td>
<td>1529, 1530</td>
</tr>
<tr>
<td>897, 908, 913</td>
<td>1211, 1212</td>
<td>1531, 1532</td>
</tr>
<tr>
<td>914, 919, 920</td>
<td>1213, 1214</td>
<td>1533, 1534</td>
</tr>
<tr>
<td>921, 922, 927</td>
<td>1215, 1216</td>
<td>1536, 1544</td>
</tr>
<tr>
<td>928, 929, 930</td>
<td>1217, 1218</td>
<td>1546, 1554</td>
</tr>
<tr>
<td>931, 932, 943</td>
<td>1219, 1220</td>
<td>1555, 1557</td>
</tr>
<tr>
<td>948, 949, 960</td>
<td>1221, 1222</td>
<td>1565, 1567</td>
</tr>
<tr>
<td>965, 966, 967</td>
<td>1223, 1224</td>
<td>1575, 1576</td>
</tr>
<tr>
<td>968, 969, 970</td>
<td>1226, 1234</td>
<td>1578, 1586</td>
</tr>
<tr>
<td>971, 972, 983</td>
<td>1236, 1244</td>
<td>1588, 1596</td>
</tr>
<tr>
<td>988, 989, 1000</td>
<td>1245, 1247</td>
<td>1597, 1599</td>
</tr>
<tr>
<td>1005, 1006</td>
<td>1255, 1257</td>
<td>1607, 1609</td>
</tr>
<tr>
<td>1015, 1037</td>
<td>1265, 1266</td>
<td>1617, 1618</td>
</tr>
<tr>
<td>1038, 1041</td>
<td>1268, 1276</td>
<td>1620, 1628</td>
</tr>
<tr>
<td>1052, 1063</td>
<td>1278, 1286</td>
<td>1630, 1638</td>
</tr>
<tr>
<td>1074, 1085</td>
<td>1287, 1289</td>
<td>1639, 1641</td>
</tr>
<tr>
<td>1094, 1103</td>
<td>1297, 1299</td>
<td>1649, 1651</td>
</tr>
<tr>
<td>1112, 1121</td>
<td>1307, 1308</td>
<td>1659, 1660</td>
</tr>
<tr>
<td>1134, 1135</td>
<td>1310, 1318</td>
<td>1662, 1670</td>
</tr>
<tr>
<td>1136, 1137</td>
<td>1320, 1328</td>
<td>1672, 1680</td>
</tr>
<tr>
<td>1138, 1139</td>
<td>1329, 1331</td>
<td>1681, 1683</td>
</tr>
<tr>
<td>1140, 1141</td>
<td>1339, 1341</td>
<td>1691, 1693</td>
</tr>
<tr>
<td>Year 1</td>
<td>Year 2</td>
<td>Year 3</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>1701</td>
<td>1702</td>
<td>2018</td>
</tr>
<tr>
<td>1704</td>
<td>1712</td>
<td>2020</td>
</tr>
<tr>
<td>1714</td>
<td>1722</td>
<td>2022</td>
</tr>
<tr>
<td>1723</td>
<td>1725</td>
<td>2024</td>
</tr>
<tr>
<td>1733</td>
<td>1735</td>
<td>2026</td>
</tr>
<tr>
<td>1743</td>
<td>1744</td>
<td>2028</td>
</tr>
<tr>
<td>1746</td>
<td>1755</td>
<td>2030</td>
</tr>
<tr>
<td>1757</td>
<td>1765</td>
<td>2033</td>
</tr>
<tr>
<td>1766</td>
<td>1768</td>
<td>2043</td>
</tr>
<tr>
<td>1776</td>
<td>1778</td>
<td>2052</td>
</tr>
<tr>
<td>1786</td>
<td>1787</td>
<td>2063</td>
</tr>
<tr>
<td>1789</td>
<td>1797</td>
<td>2074</td>
</tr>
<tr>
<td>1799</td>
<td>1807</td>
<td>2077</td>
</tr>
<tr>
<td>1808</td>
<td>1810</td>
<td>2087</td>
</tr>
<tr>
<td>1818</td>
<td>1820</td>
<td>2096</td>
</tr>
<tr>
<td>1828</td>
<td>1829</td>
<td>2106</td>
</tr>
<tr>
<td>1831</td>
<td>1839</td>
<td>2116</td>
</tr>
<tr>
<td>1841</td>
<td>1849</td>
<td>2119</td>
</tr>
<tr>
<td>1850</td>
<td>1852</td>
<td>2129</td>
</tr>
<tr>
<td>1860</td>
<td>1862</td>
<td>2138</td>
</tr>
<tr>
<td>1870</td>
<td>1871</td>
<td>2148</td>
</tr>
<tr>
<td>1873</td>
<td>1881</td>
<td>2158</td>
</tr>
<tr>
<td>1883</td>
<td>1891</td>
<td>2161</td>
</tr>
<tr>
<td>1892</td>
<td>1894</td>
<td>2171</td>
</tr>
<tr>
<td>1902</td>
<td>1904</td>
<td>2180</td>
</tr>
<tr>
<td>1912</td>
<td>1913</td>
<td>2190</td>
</tr>
<tr>
<td>1914</td>
<td>1915</td>
<td>2200</td>
</tr>
<tr>
<td>1916</td>
<td>1917</td>
<td>2202</td>
</tr>
<tr>
<td>1918</td>
<td>1919</td>
<td>2204</td>
</tr>
<tr>
<td>1920</td>
<td>1921</td>
<td>2206</td>
</tr>
<tr>
<td>1922</td>
<td>1923</td>
<td>2208</td>
</tr>
<tr>
<td>1924</td>
<td>1925</td>
<td>2210</td>
</tr>
<tr>
<td>1933</td>
<td>1934</td>
<td>2212</td>
</tr>
<tr>
<td>1935</td>
<td>1936</td>
<td>2214</td>
</tr>
<tr>
<td>1944</td>
<td>1945</td>
<td>2216</td>
</tr>
<tr>
<td>1946</td>
<td>1947</td>
<td>2218</td>
</tr>
<tr>
<td>1955</td>
<td>1956</td>
<td>2220</td>
</tr>
<tr>
<td>1957</td>
<td>1958</td>
<td>2222</td>
</tr>
<tr>
<td>1966</td>
<td>1967</td>
<td>2224</td>
</tr>
<tr>
<td>1968</td>
<td>1969</td>
<td>2226</td>
</tr>
<tr>
<td>1977</td>
<td>1978</td>
<td>2228</td>
</tr>
<tr>
<td>1979</td>
<td>1980</td>
<td>2230</td>
</tr>
<tr>
<td>1988</td>
<td>1989</td>
<td>2232</td>
</tr>
<tr>
<td>1990</td>
<td>1991</td>
<td>2234</td>
</tr>
<tr>
<td>1999</td>
<td>2000</td>
<td>2236</td>
</tr>
<tr>
<td>2001</td>
<td>2002</td>
<td>2238</td>
</tr>
<tr>
<td>2010</td>
<td>2011</td>
<td>2240</td>
</tr>
<tr>
<td>2012</td>
<td>2013</td>
<td>2242</td>
</tr>
<tr>
<td>2014</td>
<td>2015</td>
<td>2244</td>
</tr>
<tr>
<td>2016</td>
<td>2017</td>
<td>2246</td>
</tr>
<tr>
<td>Page Numbers</td>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>2408-2409</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2411-2413</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2414-2415</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2416-2417</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2418-2419</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2420-2421</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2422-2423</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2424-2425</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2426-2427</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2428-2429</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2430-2431</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2432-2433</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2434-2435</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2436-2437</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2438-2439</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2440-2441</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2442-2443</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2444-2445</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2446-2447</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2448-2449</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2450-2451</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2459-2460</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2461-2462</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2470-2471</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2472-2473</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2481-2482</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2483-2484</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2492-2493</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2494-2495</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2503-2504</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2505-2506</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2514-2515</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2516-2517</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2525-2526</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2527-2528</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2536-2537</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2538-2539</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2540-2541</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2542-2543</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2544-2545</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2546-2547</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2548-2549</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2550-2551</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2552-2553</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2554-2555</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2556-2557</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2558-2559</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2561-2570</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2571-2572</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2574-2582</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2583-2584</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2739-2740</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2741-2742</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2743-2744</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2745-2746</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2747-2748</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2749-2750</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2751-2752</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2753-2754</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2755-2756</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2757-2758</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2759-2760</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2761-2762</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2763-2764</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2765-2766</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2767-2768</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2769-2770</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2771-2772</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2773-2774</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2775-2776</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2777-2778</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2779-2780</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2781</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\newlength</td>
<td>27, 1175</td>
<td></td>
</tr>
<tr>
<td>\noSidebar</td>
<td>784, 788, 797,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>806, 815, 826,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>845, 882, 901,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>936, 953, 976, 993</td>
<td></td>
</tr>
<tr>
<td>\node</td>
<td>1479</td>
<td></td>
</tr>
<tr>
<td>\noSidebar</td>
<td>42, 785</td>
<td></td>
</tr>
<tr>
<td>\overbrace</td>
<td>74, 76</td>
<td></td>
</tr>
<tr>
<td>\pgfmathsetmacro</td>
<td>424, 425, 459, 460, 494,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>495, 529, 530,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>596, 597, 775,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>776, 777, 781,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>782, 783, 789,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>790, 791, 798,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>799, 800, 807,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>808, 809, 816,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>817, 818, 827,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>828, 829, 846,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>847, 848, 883,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>884, 885, 902,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>903, 904, 937,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>938, 939, 954,</td>
<td></td>
</tr>
</tbody>
</table>
Index

955, 956, 977, 1921, 2013, \Rr 91, 121, 204, 277,
978, 979, 994, 2014, 2018, 303, 558, 726, 749
995, 996, 1007, 2019, 2023, \rrB 1184
1016, 1019, 2024, 2028, \rrb 2240
1022, 1037, 2029, 2041, \rrBa 1214
1038, 1124, 2063, 2085, \rrBap 1219
1125, 1126, 2106, 2127, \rrBc 2903
1127, 1128, 2148, 2169, \rrBcp 2906
1129, 1130, 2190, 2255, \rrBm 2397
1131, 1132, 1133 2257, 2259, \rrBmp 2402
\ProvidesPackage ... 5 2260, 2261, \rrBp 1189
                                  2262, 2389, \rrbp 2250
                                  2390, 2394, \rrBp 1204
\R 2395, 2399, \rrBap 1209
\r 2400, 2404, \rrBw 1194
\Ra 2405, 2407, \rrBwp 1199
\raisebox ... 1159, 2408, 2410, \rrCB 2443
              1161, 1163, 2412, 2539, \rrCBp 2446
              1165, 1167, 2540, 2544, \rrCD 2431
              1169, 1171, 2545, 2549, \rrCDp 2434
              1174, 1186, 2550, 2554, \rrCF 2437
              1187, 1191, 2555, 2624, \rrCFp 2440
              1192, 1196, 2626, 2629, \rrCL 2419
              1197, 1201, 2631, 2722, \rrCLp 2422
              1202, 1206, 2723, 2727, \rrCR 2413
              1207, 1211, 2728, 2732, \rrCRp 2416
              1212, 1216, 2733, 2737, 2738 \rrCU 2425
              1217, 1221, \rrC 91, 121, 2281
              1222, 1234, \rrCfiledate ... 3, 5 \rrD 1224
              1255, 1276, \rrCfileversion ... 2, 5 \rrDd 2225
              1297, 1318, \relax ... 31, 43 \rrDa 1350
              1339, 1360, \RequirePackage 8, 11, 14 \rrDap 1371
              1381, 1402, \Rlb ... 91, 121, \rrDc 2284
              1423, 1442, 204, 273, 299, \rrDcp 2284
              1461, 1482, 552, 673, 751, 917 \rrDm 2365
              1502, 1516, \Rlm ... 90, 120, \rrDmp 2376
              1517, 1521, 188, 196, 266, \rrDp 1245
              1522, 1526, 292, 542, 674, 916 \rrDp 2226
              1527, 1531, \Rlt ... 89, 119, \rrDs 1308
              1532, 1544, 180, 259, 285, \rrDsp 1329
              1565, 1586, 532, 654, 675, 915 \rrDw 1266
              1607, 1628, \Rm ... 26 \rrDwp 1287
              1649, 1670, \Rmb ... 91, 121, 204, \rrE 1392
              1691, 1712, 275, 301, 555, 750 \rrEp 1413
              1733, 1755, \Rmm ... 90, 120, 188, \rrF 1434
              1776, 1797, 196, 268, 294, 545 \rrf 2243
              1818, 1839, \Rmt ... 89, 119, 180, \rrFa 1524
              1860, 1881, 261, 287, 535, 655 \rrFap 1529
              1902, 1915, \Rp ... 25 \rrFc 2287
             1916, 1920, \rr ... 25, 1180 \rrFcp 2290
Index

rrFm ................. 2387  rrhE ........ 1392, 1935,  rrhMF ............. 2537
rrFsp ................ 2392  1944, 1946,  rrhMFp ........... 2542
rrFp ............. 1453  1955, 2355,  rrhML ........... 2471
rrfp ............. 2244  2364, 2366,  rrhMLp ........... 2482
rrFs ............. 1514  2375, 2505,  rrhMp ........... 1723, 1979,
rrFsp ........... 1519  2514, 2516, 2525  1988, 2000,
rrFw ............. 1472  rrhEp ........ 1413, 1924,  1999, 2300,
rrFwp ........... 1493  1933, 1957,  2309, 2333,
rrh ................ 25, 1180, 2558,  1966, 2344,
                 2559, 2571, 2572  2353, 2377,  2459, 2483, 2492
rrhb .............. 1184  2386, 2494,  rrhMR ............. 2449
rrhb ................ 2249, 2756  2503, 2527,  2536  rrhMRp ........... 2460
rrhBa ........... 1214  rrhF ............. 1434  rrhMU ........... 2493
rrhBap ........... 1219  rrhf ................ 2243, 2754  rrhMUp ........... 2504
rrhBc ........... 2293, 2768  rrhFa ............. 1524  rrhR ............. 1744
rrhBcp ........... 2296, 2769  rrhFap ........... 1529  rrhr .......... 2237, 2748
rrhBm ........... 2397  rrhFc ............. 2287, 2766  rrhRa ........... 1871
rrhBmp ........... 2402  rrhFcp ........... 2290, 2767  rrhRp ........... 1892
rrhBp ........... 1180  rrhFm ........... 2387  rrhRc ........... 2633, 2760
rrhbp ........... 2252, 2757  rrhFmp ........... 2392  rrhRcp ........... 2666, 2761
rrhBs ........... 1204  rrhFp ............. 1453  rrhRm ........... 2299
rrhBsp ........... 1209  rrhfp ........... 2246, 2755  rrhRsp ........... 2310
rrhBw ........... 1194  rrhFs ............. 1514  rrhRp ........... 1766
rrhBwp ........... 1199  rrhFsp ........... 1519  rrhrp ........... 2240, 2749
rrhCB ........... 2443, 2780  rrhFw ............. 1472  rrhRs ........ 1829, 2633, 2642
rrhCBp ........... 2446, 2781  rrhFwp ........... 1493  rrhRsp ........ 1850, 2644, 2653
rrhCD ........... 2431, 2776  rrhH ............. 1534  rrhRw ........... 1787
rrhCDp ........... 2434, 2777  rrhHl ............. 2231, 2746  rrhRwp ........... 1808
rrhCF ........... 2437, 2778  rrhHa ............. 1660  rrhS ........... 1913
rrhCFp ........... 2440, 2779  rrhHap ........... 1681  rrhSB ........... 2730
rrhCL ........... 2419, 2770  rrhLc ............. 2269, 2758  rrhSc ........... 2021
rrhCLp ........... 2422, 2771  rrhLcp ........... 2272, 2759  rrhSp ........... 2735
rrhCR ........... 2413, 2772  rrhLm ............. 2321  rrhSp ........... 2026
rrhCRp ........... 2416, 2773  rrhLmp ........... 2332  rrhSp ........... 2026
rrhCU ........... 2425, 2774  rrhLp ............. 1555  rrhSp ........... 2026
rrhCUp ........... 2428, 2775  rrhhp ........... 2324, 2747  rrhSD ........... 2026
rrhD ........... 1224  rrhls ........ 1618, 2655, 2664  rrhSDp ........... 2700
rrhd ........... 2225, 2752  rrhLsp ........ 1639, 2666, 2675  rrhSF ........... 2720
rrhDa ........... 1350  rrhLw ............. 1576  rrhSF ........... 2011
rrhDap ........... 1371  rrhLwp .......... 1597  rrhSfp ........... 2725
rrhDc ........... 2281, 2764  rrhM ........... 1702, 1968,  rrhSfp ........... 2016
rrhDcp ........... 2284, 2765  1977, 2001,  rrhSL ........... 2654
rrhDm ........... 2365  2010, 2311,  rrhSI ........... 1967
rrhDmp ........... 2376  2320, 2322,  rrhSlp ........... 2665
rrhDp ........... 1245  2331, 2461,  rrhSlp ........... 1978
rrhd ........... 2228, 2753  2470, 2472, 2481  rrhSp ........... 1918
rrhDs ........ 1308, 2699, 2708  rrhMB ........... 2547  rrhSr ........... 2632
rrhDsp ........ 1329, 2710, 2719  rrhMBp ........... 2552  rrhSr ........... 1989
rrhDw ........... 1266  rrhMD ........... 2515  rrhSp ........... 2643
rrhDwp ........... 1287  rrhMDp ........... 2526  rrhSr ........... 2000


Index

<table>
<thead>
<tr>
<th>Index</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>\rthSU</td>
<td>2676</td>
</tr>
<tr>
<td>\rthSu</td>
<td>1923</td>
</tr>
<tr>
<td>\rthSup</td>
<td>2687</td>
</tr>
<tr>
<td>\rthSup</td>
<td>1934</td>
</tr>
<tr>
<td>\rthTB</td>
<td>2622</td>
</tr>
<tr>
<td>\rthTBP</td>
<td>2627</td>
</tr>
<tr>
<td>\rthTD</td>
<td>2606</td>
</tr>
<tr>
<td>\rthTDp</td>
<td>2610</td>
</tr>
<tr>
<td>\rthUF</td>
<td>2614</td>
</tr>
<tr>
<td>\rthTF</td>
<td>2618</td>
</tr>
<tr>
<td>\rthTL</td>
<td>2582</td>
</tr>
<tr>
<td>\rthTLP</td>
<td>2586</td>
</tr>
<tr>
<td>\rthTR</td>
<td>2590</td>
</tr>
<tr>
<td>\rthTRP</td>
<td>2594</td>
</tr>
<tr>
<td>\rthTU</td>
<td>2598</td>
</tr>
<tr>
<td>\rthTUP</td>
<td>2602</td>
</tr>
<tr>
<td>\rths</td>
<td>2031</td>
</tr>
<tr>
<td>\rthu</td>
<td>2219, 2750</td>
</tr>
<tr>
<td>\rthuA</td>
<td>2159</td>
</tr>
<tr>
<td>\rthSap</td>
<td>2180</td>
</tr>
<tr>
<td>\rthUc</td>
<td>2275, 2762</td>
</tr>
<tr>
<td>\rthUcp</td>
<td>2278, 2763</td>
</tr>
<tr>
<td>\rthUm</td>
<td>2343</td>
</tr>
<tr>
<td>\rthUmp</td>
<td>2354</td>
</tr>
<tr>
<td>\rthUp</td>
<td>2075</td>
</tr>
<tr>
<td>\rthUp</td>
<td>2222, 2751</td>
</tr>
<tr>
<td>\rthUS</td>
<td>2117, 2138, 2677, 2686</td>
</tr>
<tr>
<td>\rthUsap</td>
<td>2148, 2158, 2688, 2697</td>
</tr>
<tr>
<td>\rthUw</td>
<td>2052</td>
</tr>
<tr>
<td>\rthUwp</td>
<td>2096</td>
</tr>
<tr>
<td>\rthx</td>
<td>2201, 2740</td>
</tr>
<tr>
<td>\rthxp</td>
<td>2204, 2741</td>
</tr>
<tr>
<td>\rthy</td>
<td>2207, 2742</td>
</tr>
<tr>
<td>\rthyp</td>
<td>2210, 2743</td>
</tr>
<tr>
<td>\rthz</td>
<td>2213, 2744</td>
</tr>
<tr>
<td>\rthzp</td>
<td>2216, 2745</td>
</tr>
<tr>
<td>\rL</td>
<td>1534</td>
</tr>
<tr>
<td>\rL</td>
<td>2231</td>
</tr>
<tr>
<td>\rLa</td>
<td>1660</td>
</tr>
<tr>
<td>\rLap</td>
<td>1681</td>
</tr>
<tr>
<td>\rLc</td>
<td>2269</td>
</tr>
<tr>
<td>\rLcp</td>
<td>2272</td>
</tr>
<tr>
<td>\rLm</td>
<td>1555</td>
</tr>
<tr>
<td>\rLm</td>
<td>2321</td>
</tr>
<tr>
<td>\rLnp</td>
<td>2332</td>
</tr>
<tr>
<td>\rLp</td>
<td>1618</td>
</tr>
<tr>
<td>\rLap</td>
<td>1630</td>
</tr>
</tbody>
</table>
Index

\RubikBm .......... 2397 \RubikDw .......... 1266 \RubikLcp .......... 2272
\RubikBmp .......... 2402 \RubikDwp .......... 1287 \RubikLm .......... 2321
\RubikEp .......... 1189 \RubikEp .......... 1392 \RubikLmp .......... 2332
\Rubikbp .......... 2254 \Rubikf .......... 1413 \Rubiklp .......... 1555
\RubikBs .......... 1204 \RubikF .......... 1434 \Rubiklp .......... 2236
\RubikEsp .......... 1200 \Rubikf .......... 2243 \Rubikls .......... 1618
\RubikBw .......... 1194 \Rubikfa .......... 1524 \Rubiklsp .......... 1630
\RubikBwp .......... 1199 \RubikFace .......... 19 \Rubiklw .......... 1576
\RubikCB .......... 2443 \RubikFaceBack .......... 20, \Rubiklwp .......... 1597
\RubikCbp .......... 2446 78, 103, 150, 162 \RubikM .......... 1702
\RubikCD .......... 2431 \RubikFaceBackAll .......... 20, \RubikMB .......... 2547
\RubikCdp .......... 2434 108, 133, 143, 171 \RubikMBp .......... 2552
\RubikCF .......... 2437 \RubikFaceDown .......... 20, \RubikMD .......... 2515
\RubikCfp .......... 2440 20, 78, 93, 148, 160 \RubikMDp .......... 2526
\RubikCL .......... 2419 \RubikFaceDownAll .......... 20, \RubikMF .......... 2537
\RubikClp .......... 2422 108, 128, 139, 169 \RubikMfp .......... 2542
\RubikCR .......... 2413 \RubikFaceFront .......... 20, \RubikML .......... 2471
\RubikCrp .......... 2416 20, 78, 83, 149, 161 \RubikMlp .......... 2482
\RubikCU .......... 2425 \RubikFaceFrontAll .......... \RubikMp .......... 1723
\Rubikcube .......... 16 \RubikCube .......... 20, \RubikMr .......... 2449
\RubikCube ........ 11, 15 108, 113, 142, 170 \RubikMBp .......... 2460
\RubikCubeGray .......... 152 \RubikFaceLeft .......... \RubikMU .......... 2493
\RubikCubeGrayAll .......... 174 20, 78, 98, 146, 158 \RubikMUp .......... 2504
\RubikCubeGrayWB .......... 164 \RubikFaceLeftAll .......... 20, \RubikR .......... 1744
\RubikCubeGrayWY .......... 154 108, 123, 140, 167 \RubikR .......... 2237
\RubikCubeGrey .......... 144 \RubikFaceRight .......... \RubikRa .......... 1871
\RubikCubeGrey .......... 19 20, 78, 88, 145, 157 \RubikRap .......... 1892
\RubikCubeGreyAll .......... 19, 29, 173 \RubikFaceRightAll .......... 20, \RubikRc .......... 2263
\RubikCubeGreyWB .......... 23, 144 20, 78, 98, 146, 158 \RubikRcp .......... 2266
\RubikCubeGreyWY .......... 23, 144 \RubikFaceUp .......... \RubikRm .......... 2299
\RubikCubeSolved .......... 21, 175 \RubikFaceUp .......... \RubikRmp .......... 2310
\RubikCubeSolved .......... 21, 175 20, 78, 78, 147, 159 \RubikRp .......... 1706
\RubikCubeSolved .......... 19 \RubikFaceUpAll .......... 20, \RubikRrp .......... 2242
\RubikCubeSolvedWB .......... 22, 175 \RubikFap .......... 1529 \RubikRsp .......... 1850
\RubikCubeSolvedWY .......... 21, 175 \RubikFc .......... 2287 \RubikRw .......... 1787
\RubikCUp .......... 2428 \RubikFcp .......... 2290 \RubikRwp .......... 1808
\Rubikd .......... 1224 \RubikFm .......... 2387 \RubikS .......... 1913
\Rubikd .......... 2224 \RubikFmp .......... 2392 \RubikSB .......... 2730
\RubikDa .......... 1350 \RubikFp .......... 1453 \RubikSb .......... 2021
\RubikDap .......... 1371 \Rubikfp .......... 2248 \RubikSBp .......... 2735
\RubikDc .......... 2281 \RubikFs .......... 1514 \RubikSbp .......... 2026
\RubikDcp .......... 2284 \RubikFsp .......... 1519 \RubikSD .......... 2698
\RubikDm .......... 2365 \RubikFw .......... 1472 \RubikSd .......... 1945
\RubikDmp .......... 2376 \RubikFwp .......... 1493 \RubikSDp .......... 2709
\RubikDp .......... 1245 \RubikL .......... 1534 \RubikSdp .......... 1956
\Rubikdp .......... 2230 \RubikLa .......... 1600 \RubikSf .......... 2720
\RubikDs .......... 1308 \RubikLap .......... 1681 \RubikSfp .......... 2725
\RubikDsp .......... 1329 \RubikLc .......... 2260 \RubikSfp .......... 2016
Index

\sidebarRubikTbackRU  \SquareFp        1453  \SquareSFp        2726, 2727, 2728
\sidebarT        646,  841, 853, 854, 855  \SquareFs        1515, 1516, 1517  \SquareSfp        2017, 2018, 2019
\SquareFsp        1520, 1521, 1522  \SquareFw        1472  \SquareSp        1918
\SquareFwp        1493  \SquareTB        2623, 2624, 2626
\SquareL        1534  \SquareTBP        2628, 2629, 2631
\SquareLa        1660  \SquareU        2031
\SquareLap        1681  \SquareUs        2159
\SquareLp        1555  \SquareUap        2180
\SquareLs        1618, 2660  \SquareUp        2075
\SquareLsp        1639, 2671  \SquareUs        2117, 2682
\SquareLw        1576  \SquareUs        2138, 2693
\SquareLwp        1597  \SquareUs        2052
\SquareMa        1702, 1793, 2006, 2316, 2371, 2510, 2521
\SquareMb        2382, 2499, 2532
\SquareM        1723, 1984, 1995, 2305
\SquareMbp        2388, 2389, 2390
\SquareMbp        2403, 2404, 2405
\SquareMf        2405, 2503, 2554, 2555
\SquareMfp        2538, 2539, 2540
\SquareMp        1723, 1984, 1995, 2305
\SquareR        1744, 2338, 2455, 2488
\SquareRa        1871
\SquareRap        1892
\SquareRb        1766
\SquareRb       1829, 1829, 1829
\SquareRbp       1850, 1850, 1850
\SquareRwp       1808
\SquareRw        1787
\SquareRu        1813
\SquareRy        1913
\SquareS        1913
\SquareSb        2731, 2732, 2733
\SquareSb       2382, 2499, 2532
\SquareSbp       2022, 2023, 2024
\SquareSp        2736, 2737, 2738
\SquareSp        2027, 2028, 2029
\SquareSp        2388, 2389, 2390
\SquareSp        2721, 2722, 2723
\SquareSp        2393, 2394, 2395
\SquareSp        2012, 2013, 2014
\textRubikCB        2780
\textRubikB     2768
\textRubikBc       2769
\textRubikBcp       2397
\textRubikBm        2402
\textRubikBp        1189
\textRubikBn        2756
\textRubikBa       1214
\textRubikBap       1219
\textRubikBc        2768
\textRubikBcp       2769
\textRubikBm        2397
\textRubikBn        2402
\textRubikBp        1189
\textRubikBn        2757
\textRubikBn        1204
\textRubikBp        1200
\textRubikBw        1194
\textRubikBw        1199
\textRubikCB        2780
Index

\[ \text{U} \]
\[ \text{\textbackslash U} \quad \text{underbrace} \quad . . . \quad 75, 77 \]
\[ \text{\textbackslash U} \quad \text{\textbackslash up} \quad . . . \quad 25 \]
\[ \text{\textbackslash u} \quad . . . \quad 25 \]
\[ \text{\textbackslash ur} \quad . . \quad 25 \]
\[ \text{\textbackslash ur} \quad 81, 111, 255, 419, \]
\[ \text{\textbackslash ur} \quad 453, 667, 684, 925 \]
\[ \text{\textbackslash ur} \quad 441, 442, 444, \]
\[ \text{\textbackslash ur} \quad 445, 448, 449, \]
\[ \text{\textbackslash urb} \quad 81, 111, 251, 415, \]
\[ \text{\textbackslash urb} \quad 248, 412, 443, 685 \]
\[ \text{\textbackslash urb} \quad 451, 452, 454, 455 \]
\[ \text{\textbackslash urm} \quad . . . \quad 80, 110, \]
\[ \text{\textbackslash urm} \quad 405, 433, 686, 722 \]
\[ \text{\textbackslash urt} \quad . . \quad 79, 109, 241, \]
\[ \text{\textbackslash urt} \quad 244, 408, 437, 704 \]
\[ \text{\textbackslash us} \quad . . \quad 27 \]
\[ \text{\textbackslash us} \quad \text{x} \quad . . \quad 28, 52, 56, \]
\[ \text{\textbackslash ult} \quad . . \quad 27 \]
\[ \text{\textbackslash ult} \quad \text{\textbackslash usefont} \quad . . \quad 24, 25 \]
\[ \text{\textbackslash ult} \quad 62, 67, 1010, 1013 \]
\[ \text{\textbackslash w} \quad . . \quad 27 \]
\[ \text{\textbackslash w} \quad \text{xcount} \quad . . \quad 53, 56, 62, 67 \]
\[ \text{\textbackslash um} \quad . . \quad 26 \]
\[ \text{\textbackslash um} \quad \text{\textbackslash ux} \quad . . \quad 424, 428, 429, \]
\[ \text{\textbackslash um} \quad 431, 432, 434, \]
\[ \text{\textbackslash um} \quad 435, 438, 439, \]
\[ \text{\textbackslash um} \quad \text{\textbackslash y} \quad . . \quad 28, 1011, 1013 \]
\[ \text{\textbackslash umm} \quad . . \quad 26 \]
\[ \text{\textbackslash umm} \quad 441, 442, 444, \]
\[ \text{\textbackslash umm} \quad 445, 448, 449, \]
\[ \text{\textbackslash umt} \quad . . \quad 28, 1012, 1013 \]
\[ \text{\textbackslash umt} \quad . . \quad 28, 1012, 1013 \]
\[ \text{\textbackslash underbrace} \quad . . \quad 75, 77 \]
\[ \text{\textbackslash uy} \quad . . \quad 425, 428, 429, \]
\[ \text{\textbackslash uy} \quad . . \quad 431, 432, 434, \]
\[ \text{\textbackslash uy} \quad . . \quad 435, 438, 439, \]
\[ \text{\textbackslash uy} \quad . . \quad 441, 442, 444, \]
\[ \text{\textbackslash uy} \quad . . \quad 445, 448, 449, \]
\[ \text{\textbackslash uy} \quad . . \quad 451, 452, 454, 455 \]
\[ \text{\textbackslash uy} \quad . . \quad 451, 452, 454, 455 \]
\[ \text{\textbackslash ux} \quad . . \quad 424, 428, 429, \]
\[ \text{\textbackslash ux} \quad 431, 432, 434, \]
\[ \text{\textbackslash ux} \quad 435, 438, 439, \]
\[ \text{\textbackslash y} \quad . . \quad 28, 1011, 1013 \]
\[ \text{\textbackslash z} \quad . . \quad 28, 1012, 1013 \]