The \texttt{karnaughmap} package\footnote{This document corresponds to \texttt{karnaughmap} v2.0, dated 2015/04/20.}

Michael Vonbun
\texttt{m.vonbun@gmail.com}

April 20, 2015

\section{Introduction}

This package uses TikZ to typeset Karnaugh maps in an easy way. Though the map size is restricted to four variables (as for v2.0), it is easier and customizable compared to \texttt{karnaugh.tex}.

\section{Installation}

Install this package like any other \LaTeX{} package.

The end of this file contains the installation file (.ins) code in case the provided ins-file got lost.

For style creation run \texttt{latex karnaughmap.ins}. For doc/manual creation run \texttt{latex karnaughmap.dtx} afterwards.

\section{Dependencies}

As of \texttt{karnaughmap} v2.0 this package depends on
\begin{itemize}
\item tikz
\item xkeyval
\item ifthen
\item xstring
\end{itemize}
4 Usage

4.1 Karnaugh Map Macros

\karnaughmap\[(key=value)\}{\langle\text{entries}\rangle}

Use this macro inside of a \texttt{tikzpicture}-environment to typeset a Karnaugh map of a corresponding truth table.

Example  Assume the truth table with inputs \(a\) and \(b\) and output \(z\) is

\[
\begin{array}{ccc}
  a & b & z \\
  0 & 0 & 0 \\
  0 & 1 & 0 \\
  1 & 0 & \ast \\
  1 & 1 & 1 \\
\end{array}
\]

Then, the corresponding Karnaugh map with default layout is:

\[
\begin{array}{c|c|c|c}
  & & & \\
  & a & \ast & z \\
 0 & 0 & 2 & 1 \\
 1 & 3 & 1 \\
\end{array}
\]

Which was typeset using:

\begin{tikzpicture}[thick]
\karnaughmap{00\ast1}
\end{tikzpicture}

To produce empty maps, there are two possibilities. First is to use the key \texttt{defaultmap} described later and omit the mandatory input, the second is by providing a scalar mandatory argument that is equal the number of cells desired, i.e. 4, 8, or 16.

\[
\begin{array}{c|c|c}
  & & \\
  & a & \ast \\
 0 & 0 & 2 \\
 1 & 3 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
  & & \\
  & a & \ast \\
 0 & 0 & 2 \\
 1 & 3 \\
\end{array}
\]

Note that it is also possible to provide the input string with blanks:

\begin{tikzpicture}[thick]
\karnaughmap{1100 0101}
\end{tikzpicture}
This is especially useful when larger maps are typeset as it increases readability of the \LaTeX code.

\karnaughmapcolorfield \karnaughmapcolorfield[(key=value)]{(map size)}{(field string)}{(color)}

Use this macro inside of a \texttt{tikzpicture}-environment to color a Karnaugh map of size \texttt{map size}. Use hexadecimal values to address a certain field of the map in the \texttt{field string}.

\begin{tikzpicture}[thick]
\karnaughmapcolorfield{2}{01}{teal!50}\
\karnaughmapcolorfield{2}{3}{violet!50}\
\karnaughmap{4}
\end{tikzpicture}

\begin{tikzpicture}[thick]
\karnaughmapcolorfield[outline]{2}{01}{teal!50}\
\karnaughmapcolorfield[outline,ultra thick]{2}{3}{violet!50}\
\karnaughmap{4}
\end{tikzpicture}

The keys for this macro allow for two different styles: \textit{filled} \texttt{fill} and \textit{outlined} \texttt{outline}. The default is filled. In addition, TikZ keys are forwarded to the \texttt{\draw} or \texttt{\fill} command, allowing a good adjustment of styles.

\begin{tikzpicture}[thick]
\karnaughmapcolorfield[outline]{2}{01}{teal!50}\
\karnaughmapcolorfield[outline,ultra thick]{2}{3}{violet!50}\
\karnaughmap{4}
\end{tikzpicture}
4.2 Style Customizations

For style customization a key-value system is provided.
\setkarnaughmap[\texttt{(key=value)}] This is the interface function for setting the keys. To restore the defaults call \setkarnaughmap.

The defaults are
\setkarnaughmap[\texttt{omitzeros, defaultmap=16, function=z, variables=abcd, dontcare=*}].

**Key-Value List** The following table is an overview over all options provided. The exact behavior and description of the keys are described below.

<table>
<thead>
<tr>
<th>Key</th>
<th>Prefix</th>
<th>Family</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>omitzeros</td>
<td>KV</td>
<td>karnaughmap</td>
<td>boolean</td>
<td>true</td>
</tr>
<tr>
<td>omitones</td>
<td>KV</td>
<td>karnaughmap</td>
<td>boolean</td>
<td>true</td>
</tr>
<tr>
<td>omitdontcares</td>
<td>KV</td>
<td>karnaughmap</td>
<td>boolean</td>
<td>true</td>
</tr>
<tr>
<td>omitbinaries</td>
<td>KV</td>
<td>karnaughmap</td>
<td>boolean</td>
<td>true</td>
</tr>
<tr>
<td>omitvariables</td>
<td>KV</td>
<td>karnaughmap</td>
<td>boolean</td>
<td>true</td>
</tr>
<tr>
<td>omitidx</td>
<td>KV</td>
<td>karnaughmap</td>
<td>boolean</td>
<td>true</td>
</tr>
<tr>
<td>binaryidx</td>
<td>KV</td>
<td>karnaughmap</td>
<td>boolean</td>
<td>true</td>
</tr>
<tr>
<td>omitnegated</td>
<td>KV</td>
<td>karnaughmap</td>
<td>boolean</td>
<td>true</td>
</tr>
<tr>
<td>outline</td>
<td>KV</td>
<td>karnaughmap</td>
<td>boolean</td>
<td>true</td>
</tr>
<tr>
<td>fill</td>
<td>KV</td>
<td>karnaughmap</td>
<td>boolean</td>
<td>true</td>
</tr>
<tr>
<td>opacity</td>
<td>KV</td>
<td>karnaughmap</td>
<td>command</td>
<td>0.5</td>
</tr>
<tr>
<td>defaultmap</td>
<td>KV</td>
<td>karnaughmap</td>
<td>command</td>
<td>16</td>
</tr>
<tr>
<td>function</td>
<td>KV</td>
<td>karnaughmap</td>
<td>command</td>
<td>z</td>
</tr>
<tr>
<td>variables</td>
<td>KV</td>
<td>karnaughmap</td>
<td>command</td>
<td>abcd</td>
</tr>
<tr>
<td>dontcare</td>
<td>KV</td>
<td>karnaughmap</td>
<td>command</td>
<td>*</td>
</tr>
<tr>
<td>color</td>
<td>KV</td>
<td>karnaughmap</td>
<td>ordinary</td>
<td>black</td>
</tr>
<tr>
<td>draw</td>
<td>KV</td>
<td>karnaughmap</td>
<td>ordinary</td>
<td>karnaughmapColor</td>
</tr>
<tr>
<td>labelcolor</td>
<td>KV</td>
<td>karnaughmap</td>
<td>ordinary</td>
<td>karnaughmapColor</td>
</tr>
<tr>
<td>xshift</td>
<td>KV</td>
<td>karnaughmap</td>
<td>command</td>
<td>0</td>
</tr>
<tr>
<td>yshift</td>
<td>KV</td>
<td>karnaughmap</td>
<td>command</td>
<td>0</td>
</tr>
</tbody>
</table>

**Selective Entry Printing** The following keys control the printing of different types of Karnaugh map cell entries.
\textbf{omitzeros} \texttt{\textbackslash omitzeros[=true/false]} Toggle between printing and omitting zero-valued entries in the Karnaugh map.

\begin{verbatim}
\karnaughmap[omitzeros=false]{00*1}
\end{verbatim}

\textbf{omitones} \texttt{\textbackslash omitones[=true/false]} Toggle between printing and omitting one-valued entries in the Karnaugh map.

\begin{verbatim}
\karnaughmap[omitzeros=false,omitones]{00*1}
\end{verbatim}

\textbf{omitdontcares} \texttt{\textbackslash omitdontcares[=true/false]} Toggle between printing and omitting don’t care entries in the Karnaugh map.

\begin{verbatim}
\karnaughmap[omitzeros=false,omitdontcares]{00*1}
\end{verbatim}

\textbf{Karnaugh Map Style} The following keys control the printing of different types of Karnaugh map labels.

\textbf{omitbinaries} \texttt{\textbackslash omitbinaries[=true/false]} Toggle between printing and omitting the binary encoding on top and left of the Karnaugh map.

\begin{verbatim}
\karnaughmap[omitbinaries]{00*1}
\end{verbatim}
omitvariables \omitvariables\==(true/false)\] Toggle between printing and omitting the variable labeling of rows and columns around the Karnaugh map.

\karnaughmap[omitvariables]\{00*1\}

omitidx \omitidx\==(true/false)\] Toggle between printing and omitting index values besides the cell entries in the Karnaugh map.

\karnaughmap[omitidx]\{00*1\}

binaryidx \binaryidx\==(true/false)\] Toggle between decimal and binary indices.

\karnaughmap[binaryidx]\{00*1\}

omitnegated \omitnegated\==(true/false)\] Toggle between typesetting the negated in addition to the non-negated variable labels.
Map Customizations  The following keys can be used to customize the Karnaugh maps in various ways.

**defaultmap**  defaultmap[(=16)] Set the default map size (4, 8, or 16), i.e. this map size is used if the mandatory argument of \texttt{\karnaughmap{}} is empty. Thus, empty maps can be produced.

\begin{align*}
\texttt{\karnaughmap[defaultmap=4]{}}, \texttt{\karnaughmap[defaultmap=8]{}},
\end{align*}

Note that the same functionality can be achieved by providing a scalar mandatory input

\begin{align*}
\texttt{\karnaughmap{4}}, \texttt{\karnaughmap[defaultmap=8]{}},
\end{align*}

**function**  function[(=z)] Set the function label used to label the Karnaugh map.

\begin{align*}
\texttt{\karnaughmap[function=\{f(a,b)\}]{}{00*1}}
\end{align*}
Set the variable names used in the Karnaugh map. Note that you have to provide a single string of variables or variables embraced by curly braces.

\karnaughmap\{variables\{abcd\}\{00*1\}\}

\karnaughmap\{variables\{kl\}\{00*1\}\}

\karnaughmap\{variables\{s_2\{s_1\{s_0\{a\}\}\}\}\{01** 10** 0101 **00\}\}
\textbf{dontcare} \texttt{dontcare[\{=\*\}]} Set the don’t care character. This is only needed if you provide the \texttt{omitdontcares} key and use don’t care characters other than \(*\) in the \texttt{\karnaughmap(x)} inputstring.

\texttt{\karnaughmap[dontcare=X,omitdontcares,omitzeros=false]{00X1}}

\begin{center}
\begin{tabular}{c|cc}
\textbf{a} & 0 & 1 \\
\hline
0 & 0 & 2 \\
1 & 0 & 3 \\
\end{tabular}
\end{center}

\textbf{color} \texttt{color[\{=\textcolor{black}\}]} Set the color of the \textit{entire} Karnaugh map, including labels, entries, etc.

\texttt{\karnaughmap[color=teal]{00\*1}}

\begin{center}
\begin{tabular}{c|cc}
\textbf{a} & 0 & 1 \\
\hline
0 & 0 & \* \\
1 & 3 & 1 \\
\end{tabular}
\end{center}

\textbf{draw} \texttt{draw[\{=\texttt{same\ as\ color}\}]} Set the color of the Karnaugh map \textit{only}, i.e. the lines that are drawn. The labels are not affected.

\texttt{\karnaughmap[draw=teal]{00\*1}}

\begin{center}
\begin{tabular}{c|cc}
\textbf{a} & 0 & 1 \\
\hline
0 & 0 & \* \\
1 & 3 & 1 \\
\end{tabular}
\end{center}

\textbf{labelcolor} \texttt{labelcolor[\{=\texttt{same\ as\ color}\}]} Set the color of the labels of the Karnaugh map \textit{only}. The lines are not affected.

\texttt{\karnaughmap[labelcolor=teal]{00\*1}}
Multiple Karnaugh Maps in one TikZ Picture

To typeset more than one Karnaugh map into one TikZ Picture, use \texttt{xshift} and \texttt{yshift}, just as usual in TikZ.

\texttt{xshift}

\texttt{xshift[\{\texttt{=}0\}\}] Shift the Karnaugh map in x-direction within the TikZ picture.

\begin{verbatim}
\karnaughmap{4}\karnaughmap[\texttt{xshift=5cm}]{4}
\end{verbatim}

\begin{figure}
\centering
\begin{tikzpicture}
\karnaughmap{4}
\karnaughmap[\texttt{xshift=5cm}]{4}
\end{tikzpicture}
\end{figure}

\texttt{yshift}

\texttt{yshift[\{\texttt{=}0\}\}] Shift the Karnaugh map in x-direction within the TikZ picture.

\begin{verbatim}
\karnaughmap{4}\karnaughmap[\texttt{yshift=5cm}]{4}
\end{verbatim}

\begin{figure}
\centering
\begin{tikzpicture}
\karnaughmap{4}
\karnaughmap[\texttt{yshift=5cm}]{4}
\end{tikzpicture}
\end{figure}
5 Examples

5.1 Basic Usage Examples:

\karnaughmap{defaultmap=16,binaryidx,omitnegated=false}{f(a,b,c,d):
\begin{tabular}{cccc|c}
| a | b | c | d | \hline
| 00 | 01 | 11 | 10 |
\end{tabular}
}

\karnaughmap{defaultmap=4,binaryidx,omitnegated=false,color=teal}{z:
\begin{tabular}{c|cc}
| a | b | \hline
| 0 | 1 |
\end{tabular}
}

\karnaughmap{4}{z:
\begin{tabular}{c|cc}
| a | b | \hline
| 0 | 2 |
\end{tabular}
}

5.2 Field coloring

\begin{tikzpicture}[thick]
\foreach \x [count=\xi] in {0,1,2,3} {%
  \pgfmathparse{-20+30*\xi}%
\end{tikzpicture}
6 Field Allocation Reference

This package provides maps with full control up to 4 variables, which is equivalent to a map size of 16 fields. For bigger maps, the package provides just basic typesetting, i.e. just the column and row encoding and the, possibly filled matrix. However, the biggest map supported yet has 256 fields.

The reference can be read as follows: the first row contains the value of the rows, the columns are then likewise enumerated. By this procedure, you can check if everything went right (debugging) and got an easy reference.

6.1 2 Variables – 4 Fields

\begin{tikzpicture}[thick]
\karnaughmap[variables=ab]{o1 11 }
\end{tikzpicture}
6.2 3 Variables – 8 Fields
\begin{tikzpicture}[thick]
\karnaughmap[variables=abc]{o1 11 21 31 }
\end{tikzpicture}

6.3 4 Variables – 16 Fields
\begin{tikzpicture}[thick]
\karnaughmap[variables=abcd]{o123 1123 2123 3123 }
\end{tikzpicture}
6.4 5 Variables – 32 Fields

\begin{tikzpicture}[thick]
\karnaughmap[variables=abcde]{0000 0001 0010 0011 0100 0101 0110 1111}
\end{tikzpicture}

\begin{tabular}{cccccccc}
\hline
\textit{abc} & 00 & 001 & 011 & 010 & 110 & 111 & 101 & 100 \\
\hline
00 & o & 1 & 3 & 2 & 6 & 7 & 5 & 4 \\
01 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
11 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
10 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
\hline
\end{tabular}

6.5 6 Variables – 64 Fields

\begin{tikzpicture}[thick]
\karnaughmap[variables=abcdef]{01234567 11234567 21234567 31234567 41234567 51234567 61234567 71234567}
\end{tikzpicture}
### 6.6 7 Variables – 128 Fields

\begin{verbatim}
\begin{tikzpicture}[thick]
karnaughmap[variables=abcdefg]{o1234567 11234567 21234567 31234567 41234567 51234567 61234567 71234567 81234567 91234567 a1234567 b1234567 c1234567 d1234567 e1234567 f1234567}
\end{tikzpicture}
\end{verbatim}
### 6.7 8 Variables – 256 Fields

\begin{tikzpicture}[thick]
\karnaughmap[variables=abcdefg]{o123456789abcdef 1123456789abcdef 2123456789abcdef 3123456789abcdef 4123456789abcdef 5123456789abcdef 6123456789abcdef 7123456789abcdef 8123456789abcdef 9123456789abcdef 10123456789abcdef 11123456789abcdef 12123456789abcdef 13123456789abcdef 14123456789abcdef 15123456789abcdef 16123456789abcdef 17123456789abcdef 18123456789abcdef 19123456789abcdef 20123456789abcdef 21123456789abcdef 22123456789abcdef 23123456789abcdef 24123456789abcdef 25123456789abcdef 26123456789abcdef 27123456789abcdef 28123456789abcdef 29123456789abcdef 30123456789abcdef 31123456789abcdef}
\end{tikzpicture}
<table>
<thead>
<tr>
<th></th>
<th>0000</th>
<th>0001</th>
<th>0010</th>
<th>0100</th>
<th>0101</th>
<th>1100</th>
<th>1101</th>
<th>1110</th>
<th>1111</th>
<th>1010</th>
<th>1011</th>
<th>1001</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>o</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>c</td>
<td>d</td>
<td>f</td>
<td>e</td>
<td>a</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>1100</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>c</td>
</tr>
<tr>
<td>1101</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>d</td>
</tr>
<tr>
<td>1111</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>1110</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>e</td>
<td>e</td>
</tr>
<tr>
<td>1010</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>1011</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td>1001</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>
7 Macros for Internal Usage

These macros are not intended to be used by the user, rather these are auxiliary functions invoked by the main macro $\texttt{karnaughmap}(x)$.
Nevertheless, for documentation these macros are described here briefly.

8 Implementation

8.1 Load Dependencies

1 \RequirePackage{tikz}
2 \RequirePackage{xkeyval}
3 \RequirePackage{ifthen}
4 \RequirePackage{xstring}

8.2 Allocate Counters

5 \newcounter{karnaughmapIdxCounter}
6 \newcounter{karnaughmapStrCounter}
7 \setcounter{karnaughmapIdxCounter}{0}
8 \setcounter{karnaughmapStrCounter}{0}

8.3 Key-Val Interface

Switches  Simple switches are used to control basic layout options of the Karnaugh map.

These boolean switches control if specific entries – i.e. zeros, ones, and don’t cares – are printed or not (omitted).

9 \define@boolkeys{karnaughmap}{omitzeros, omitones, omitdontcares}[true]

These boolean switches control the layout of the Karnaugh map. The omitbinaries switch is used to control if binary encoding is typeset left and on top of the Karnaugh map. The omitvariables switch is used to control if the Karnaugh map is labeled with the variables. The omitidx switch controls the typesetting of decimal indices within the cells of the Karnaugh map. The binaryidx switch controls whether the indices are decimal or binary. The omitnegated switch controls whether the negated variable labels are typeset as well.

10 \define@boolkeys{karnaughmap}{omitbinaries, omitvariables, omitidx, binaryidx, omitnegated}[true]

These boolean switches control the behavior of the highlighting function.

11 \define@boolkey{karnaughmap}{outline}[true]{{setkeys{karnaughmap}{fill=false}}}
12 \define@boolkey{karnaughmap}{fill}[true]
13 \define@cmdkey{karnaughmap}{opacity}[0.5]{}

Customizations  In contrast to the boolean switches these keys are used to customize the Karnaugh map.
This key controls the default size of the Karnaugh map. It is set to 16, i.e. a 4
variable map is typeset.
\define@cmdkey{karnaughmap}{defaultmap}[16]{}

This key controls the Karnaugh map function label. The default is 'z'.
\define@cmdkey{karnaughmap}{function}[z]{}

This key controls the variables. Provide the variables as a string. Maybe more
control over variables will be added in a later version. The default is 'abcd'
\define@cmdkey{karnaughmap}{variables}[abcd]{}

Define the standard "Don't Care" character. The default is '*'.
\define@cmdkey{karnaughmap}{dontcare}[*]{}

Define the color. For desired functionality both the draw and color have to be
set.
\define@key{karnaughmap}{color}[black]{
\colorlet{karnaughmapColor}{#1}\n\colorlet{karnaughmapMapColor}{#1}}

Define the draw color, i.e. the color of the map itself. Default is the same as
color.
\define@key{karnaughmap}{draw}[karnaughmapColor]{\colorlet{karnaughmapMapColor}{#1}}

Define the label color. By setting the color only, the draw remains unchanged.
\define@key{karnaughmap}{labelcolor}[karnaughmapColor]{\colorlet{karnaughmapColor}{#1}}

Define the shifts.
\define@cmdkeys{karnaughmap}{xshift,yshift}[0]

Set Key Defaults  The default is a map with full information, i.e. binary and
variable labels as well as decimal indices. The zeros are omitted by default.
\setkarnaughmap
Simple interface function.
\newcommand{\setkarnaughmap}[1][]{\ifthenelse{\equal{#1}{}\{}%  Change the default behavior by uncommenting the specific key.
\setkeys{karnaughmap}{%omitones, omitdontcares, omitbinaries, omitvariables, omitidx, binaryidx
omitzeros, omitnegated}\setkeys{karnaughmap}{defaultmap, function, variables, dontcare, color, draw, labelcolor, xshift,yshift}\setkeys{karnaughmap}{#1}}

If the optional input is provided set the keys accordingly.
Assign the defaults during style file call.

33 \setkarnaughmap

8.4 Auxiliary Functions

These function are called by the main function to facilitate code execution.

\karnaughmapPrintIndex \karnaughmapPrintIndex[\langle numBits \rangle]

This macro typesets the indices inside the cells. If the switch binaryidx is true binary instead of decimal indices are typeset. The number provided as optional argument is used to determine the number of bits used.

Every time the macro is called, it prints the counter value of counter karnaughmapIdxCounter and increments the counter thereafter.

The size of the indices is very small (scriptsize).

34 \newcommand{\karnaughmapPrintIndex}[1][4]{% 35 \scriptsize% 36 \ifKV@karnaughmap@binaryidx

Typeset binary indices.

37 \ifthenelse{\equal{#1}{2}}{% 38 \ifcase\value{karnaughmapIdxCounter} 39 00\or 01\or 10\or 11 40 \fi% 41 \} % 42 \ifthenelse{\equal{#1}{3}}{% 43 \ifcase\value{karnaughmapIdxCounter} 44 000\or 001\or 010\or 011\or 100\or 101\or 110\or 111 45 \fi% 46 \} % 47 \ifthenelse{\equal{#1}{4}}{% 48 \ifcase\value{karnaughmapIdxCounter} 49 0000\or 0001\or 0010\or 0011\or 0100\or 0101\or 0110\or 0111\or 1000\or 1001\or 1010\or 1011\or 1100\or 1101\or 1110\or 1111 50 \fi% 51 \} % 52 \else

Typeset decimal indices.

53 $\text{the$karnaughmapIdxCounter$}$% 54 \fi% 55 \stepcounter{karnaughmapIdxCounter}%

\karnaughmapPrintValue \karnaughmapPrintValue[\langle inputstring \rangle]

This macro typesets the entries of the Karnaugh map. Every time the macro is called, it increments the counter karnaughmapStrCounter and prints the character at position of the counter value of its mandatory input.

58 \newcommand{\karnaughmapPrintValue}[2][1]{% 59 \stepcounter{karnaughmapStrCounter} % 60 \StrChar{#2}{$\text{the$karnaughmapStrCounter$}$}}
This macro is used to provide the positions of the cells of the map, as they are not straightforward (e.g. order is 0-1-3-2 for first column of 4x4 maps). It takes 5 mandatory and one optional arguments as specified above.

By default, it evaluates \texttt{function} as contents of a node at the top left of each cell of a Karnaugh map of size \texttt{size} with \texttt{rowsize} rows. By providing the bias \((x\text{-} and \, y\text{-}offset)\), the provided point is shifted accordingly. The alignment of the node is controlled by the optional argument.

\newcommand{\karnaughmapPrintCellContents}{\langle node-position \rangle \langle x\text{-offset} \rangle \langle y\text{-offset} \rangle \langle size \rangle \langle rowsize \rangle \langle function \rangle}

If map is 2x2, the order is pretty simple:
\begin{verbatim}
0 2
1 3
\end{verbatim}

If map is 2x4, the order is more complex:
\begin{verbatim}
0 2 6 4
1 3 7 5
\end{verbatim}

If map is 4x4, the same ordering as before is present in both x- and y-direction:
\begin{verbatim}
0 4 12 8
1 5 13 9
3 7 15 11
2 6 14 10
\end{verbatim}
If map is 4x8, the ordering as before has to be extended in x-direction, but from the middle in reverse order:
0 4 12 8 24 28 20 16
1 5 13 9 25 29 21 17
3 7 15 11 27 31 23 29
2 6 14 10 26 30 22 18

If map is 8x8, the ordering as before has to be extended in y-direction, too.
0 8 24 16 48 56 40 32
1 9 25 17 49 57 41 33
If map is 8x16, the ordering is:

0 8 24 16 48 56 40 32
1 9 25 17 49 57 41 33
3 11 27 19 51 59 43 35
2 10 26 18 50 58 42 34
6 14 30 22 54 62 46 38
7 15 31 23 55 63 47 39
5 13 29 21 53 61 45 37
4 12 28 20 52 60 44 36

\def\karnaughmapPCCColumnSpecifier{%
  ++( 0,-1) node[#1] {#6}
  ++( 0,-2) node[#1] {#6}
  ++( 0,+1) node[#1] {#6}
  ++( 0,-5) node[#1] {#6}
  ++( 0,+1) node[#1] {#6}
  ++( 0,+2) node[#1] {#6}
  ++( 0,-1) node[#1] {#6}
}{}

\ifthenelse{\equal{#4}{128}}{%
  \def\karnaughmapPCCColumnSpecifier{%
    $(#2,#3) node[#1] {#6}$
    \karnaughmapPCCColumnSpecifier
    ++(+1,+5) node[#1] {#6}
    \karnaughmapPCCColumnSpecifier
  }{%
    \karnaughmapPCCColumnSpecifier
  }
}\%
If map is 16x16, the ordering is:
0 8 24 16 48 56 40 32
1 9 25 17 49 57 41 33
3 11 27 19 51 59 43 35
2 10 26 18 50 58 42 34
6 14 30 22 54 62 46 38
7 15 31 23 55 63 47 39
5 13 29 21 53 61 45 37
\karnaughmapHighlightField  Macro that defines the highlighting. \karnaughmapHighlightField[(key=value)]{(coordinates)}{(color name)}

\begin{verbatim}
255 \newcommand{\karnaughmapHighlightField}[3][]{\% 
256 \ifKV@karnaughmap@fill 
257 \path[fill=#3,opacity=\cmdKV@karnaughmap@opacity,#1] (#2) rectangle ++(1,1);\% 
258 \fi 
259 \ifKV@karnaughmap@outline 
260 \draw[#3,#1] (#2) ++ (0.1,0.1) rectangle ++(0.8,0.8);\% 
261 \fi 
262 \}
\end{verbatim}

\karnaughmapShadeMapfieldTWO  Macro to shade square two input karnaugh map fields. \karnaughmapShadeMapfieldTWO[(key=value)]{(decimal field number)}{(color name)}

- Optional macro is used as key-val interface for option processing, mandatory ones specify the color used for highlighting and the fields that are to be highlighted as decimal integers from 1 to $2^\text{map size} - 1$.

\begin{verbatim}
263 \newcommand{\karnaughmapShadeMapfieldTWO}[3][]{\% 
264 \begin{scope}[scale=1] 
265 \ifthenelse{\equal{#2}{0}}{\karnaughmapHighlightField[#1]{0,1}{#3}}{}\% 
266 \ifthenelse{\equal{#2}{1}}{\karnaughmapHighlightField[#1]{0,0}{#3}}{}\% 
267 \ifthenelse{\equal{#2}{2}}{\karnaughmapHighlightField[#1]{1,1}{#3}}{}\% 
268 \ifthenelse{\equal{#2}{3}}{\karnaughmapHighlightField[#1]{1,0}{#3}}{}\% 
269 \end{scope} 
270 \}
\end{verbatim}

\karnaughmapShadeMapfieldTHREE  Macro to shade three input karnaugh map fields. \karnaughmapShadeMapfieldTHREE[(key=value)]{(decimal field number)}{(color name)}

- Optional macro is used as key-val interface for option processing, mandatory ones specify the color used for highlighting and the fields that are to be highlighted as decimal integers from 1 to $2^\text{map size} - 1$.

\begin{verbatim}
271 \newcommand{\karnaughmapShadeMapfieldTHREE}[3][]{\% 
272 \begin{scope}[scale=1] 
273 \ifthenelse{\equal{#2}{0}}{\karnaughmapHighlightField[#1]{0,1}{#3}}{}\% 
274 \ifthenelse{\equal{#2}{1}}{\karnaughmapHighlightField[#1]{0,0}{#3}}{}\% 
275 \ifthenelse{\equal{#2}{2}}{\karnaughmapHighlightField[#1]{1,1}{#3}}{}\% 
276 \ifthenelse{\equal{#2}{3}}{\karnaughmapHighlightField[#1]{1,0}{#3}}{}\% 
277 \ifthenelse{\equal{#2}{4}}{\karnaughmapHighlightField[#1]{3,1}{#3}}{}\% 
278 \ifthenelse{\equal{#2}{5}}{\karnaughmapHighlightField[#1]{3,0}{#3}}{}\% 
279 \end{scope} 
280 \}
\end{verbatim}
Macro to shade square four input karnaugh map fields.

\karnaughmapShadeMapfieldFOUR\[(key=value)]\{(decimal field number)\}\{(color name)\}

Optional macro is used as key-val interface for option processing, mandatory ones specify the color used for highlighting and the fields that are to be highlighted as decimal integers from 1 to \(2^{\text{map size}} - 1\).

Here come the main functions to typeset the map and to mark certain cells (not implemented yet).

\karnaughmap

The main macro takes one mandatory and one optional argument:

\((\{key=value\} \{(inputstring)\})\)

The optional is used as key-val interface, the mandatory is empty or holds the entry-pattern (inputstring) according to the corresponding ordered truth table of the desired logic function.
If the optional input is not empty, assign the key-val.

\ifthenelse{\equal{#1}{}}{}{\setkeys{karnaughmap}{#1}}

For easy access store the mandatory argument as variable \karnaughmapCellEntries.

\def\karnaughmapCellEntries{#2}%
\% \end{macrocode}
\%
\% Check length of mandatory argument and store the result in |\karnaughmapSize|.
\%
\% If the mandatory argument is empty,
\% use the default map size provided by the key-val interface (|defaultmap| key).
\%
\% \begin{macrocode}
\% \ifthenelse{\equal{#2}{}}{%
\def\karnaughmapSize{\cmdKV@karnaughmap@defaultmap}
\% }{%
Else, determine the length of the input string to determine the map size.
Before calculating the string length, remove any blanks in the input string.
Usage of blanks in the input string can increase readability, e.g. when groups of
four are provided.
\IfSubStr{\karnaughmapCellEntries}{ }{%
\StrDel[0]{\karnaughmapCellEntries}{ }{\karnaughmapCellEntries}
\% }{}

Now, calculate the string length. This is the old version.
\StrLen{#2}{\karnaughmapSize}%
\StrLen{\karnaughmapCellEntries}{\karnaughmapSize}%

If the input is a scalar number, i.e. it is $\geq 1000$ (decimal), it is treated as
number and is used to determine the map size. In this case adapt the variable
\karnaughmapCellEntries accordingly.
\ifthenelse{\equal{\karnaughmapSize}{1}}{
\% \ifthenelse{\karnaughmapSize < 4}{%}
\def{\karnaughmapSize}{#2}{\def\karnaughmapCellEntries{}}{%}
\% }{}

Parse the input string and omit printing of '0', '1', or '* entries if the switches
say so.
\ifKV@karnaughmap@omitzeros
\StrSubstitute[0]{\karnaughmapCellEntries}{0}{ }{\karnaughmapCellEntries}
\fi
\ifKV@karnaughmap@omitones
\StrSubstitute[0]{\karnaughmapCellEntries}{1}{ }{\karnaughmapCellEntries}
\fi
\ifKV@karnaughmap@omitdontcares
\StrSubstitute[0]{\karnaughmapCellEntries}{\cmdKV@karnaughmap@dontcare}{ }{\karnaughmapCellEntries}
\fi

Initialize internal counters.
\setcounter{karnaughmapIdxCounter}{0}
\setcounter{karnaughmapStrCounter}{0}

Extract the map variables and alloc row and col sizes. In addition, store the
number of variables in variable \karnaughmapNumVar.
The bigger karnaugh maps, i.e. 32, 64, 128 and 512, lack finer control possibilities. Rather, the variables are set such that they can be used as map descriptors, but not for some fancy stuff like indicating the variable blocks. This means, that we force to omit printing of the the indices (omitidx=true) and we force to omit printing of the variables. For the variable macros to work, we set the B and D macro to empty and put the remaining variables in the A for column and C for row variables.

The bigger karnaugh maps, i.e. 32, 64, 128 and 512, lack finer control possibilities. Rather, the variables are set such that they can be used as map descriptors, but not for some fancy stuff like indicating the variable blocks. This means, that we force to omit printing of the the indices (omitidx=true) and we force to omit printing of the variables. For the variable macros to work, we set the B and D macro to empty and put the remaining variables in the A for column and C for row variables.
Basic table  Typeset the basic table.  

Embed the draw commands in a scope environment for style control.

\begin{scope}[xshift=\cmdKV@karnaughmap@xshift,  
yshift=\cmdKV@karnaughmap@yshift,  
color=karnaughmapColor,  
draw=karnaughmapMapColor ]

Draw table boundaries.
\draw (0,0) rectangle (\karnaughmapNumCol, \karnaughmapNumRow);

Draw column seperators.
\foreach \x in \{1,2,...,\karnaughmapNumCol\} {%  
\ifthenelse{\equal{\x}{1}}{}{%  
\draw (\x-1,0) -- ++(0,\karnaughmapNumRow);%  
}%  
}%

Draw row seperators.
\foreach \x in \{1,2,...,\karnaughmapNumRow\} {%  
\ifthenelse{\equal{\x}{1}}{}{%  
\draw (0,\x-1) -- ++(\karnaughmapNumCol,0);%  
}%  
}%
Draw map labels; Diagonal line and anchor first.
\draw (0,\karnaughmapNumRow) -- ++(-0.4,0.4) coordinate(identifier)
-- ++(-0.4,0.4)

Function label at the top of the map, if the function label is non-empty the
label is followed by a colon.
\node[yshift=7.5,xshift=-3.5,right] %
  \ifthenelse{\equal{\cmdKV@karnaughmap@function}{}}{}{$\cmdKV@karnaughmap@function\colon$};

Variable label right and below the diagonal.
\path (identifier) node[above right, xshift=-2.5] {$\karnaughmapVarLabelA\karnaughmapVarLabelB$};
\path (identifier) node[below left, yshift=0.5] {$\karnaughmapVarLabelC\karnaughmapVarLabelD$};

Typeset binaries if omitbinaries switch is false;
\ifKV@karnaughmap@omitbinaries
\else
  \ifthenelse{\equal{\karnaughmapNumCol}{2}}{%\foreach \pos/\lab in {0.5/0, 1.5/1} {%
    \path (\pos, \karnaughmapNumRow) node[above] {$\lab$};%
  }%\}
  \ifthenelse{\equal{\karnaughmapNumCol}{4}}{%\foreach \pos/\lab in {0.5/00, 1.5/01, 2.5/11, 3.5/10} {%
    \path (\pos, \karnaughmapNumRow) node[above] {$\lab$};%
  }%\}
  \ifthenelse{\equal{\karnaughmapNumCol}{8}}{%\foreach \pos/\lab in {0.5/000, 1.5/001, 2.5/011, 3.5/010, 4.5/110, 5.5/111, 6.5/101, 7.5/100} {%
    \path (\pos, \karnaughmapNumRow) node[above] {$\lab$};%
  }%\}
  \ifthenelse{\equal{\karnaughmapNumCol}{16}}{%\foreach \pos/\lab in {0.5/0000, 1.5/0001, 2.5/0011, 3.5/0010, 4.5/0110, 5.5/0111, 6.5/0101, 7.5/0100, 8.5/1100, 9.5/1101, 10.5/1111, 11.5/1110, 12.5/1010, 13.5/1011, 14.5/1001} {%
    \path (\pos, \karnaughmapNumRow) node[above] {$\lab$};%
  }%\}
  \ifthenelse{\equal{\karnaughmapNumCol}{32}}{%\foreach \pos/\lab in {0.5/00000, 1.5/00001, 2.5/00011, 3.5/00010, 4.5/00110, 5.5/00111, 6.5/00101, 7.5/00100, 8.5/01100, 9.5/01101, 10.5/01111, 11.5/01110, 12.5/01010, 13.5/01011, 14.5/01001, 15.5/01000} {%
    \path (\pos, \karnaughmapNumRow) node[above] {$\lab$};%
  }%\}
  \ifthenelse{\equal{\karnaughmapNumCol}{64}}{%\foreach \pos/\lab in {0.5/000000, 1.5/000001, 2.5/000011, 3.5/000010, 4.5/000110, 5.5/000111, 6.5/000101, 7.5/000100, 8.5/001100, 9.5/001101, 10.5/001111, 11.5/001110, 12.5/001010, 13.5/001011, 14.5/001001, 15.5/001000, 16.5/011000, 17.5/011001, 18.5/011011, 19.5/011010, 20.5/011111, 21.5/011110, 22.5/011101, 23.5/011100, 24.5/010110, 25.5/010111, 26.5/010101, 27.5/010100, 28.5/010011, 29.5/010010, 30.5/010001, 31.5/010000} {%
    \path (\pos, \karnaughmapNumRow) node[above] {$\lab$};%
  }%\}
\end{tikzpicture}

2 rows: anchors are left of the first column with coordinate (0, x) and in the
middle of the cells at 0.5 and 1.5.
\ifthenelse{\equal{\karnaughmapNumRow}{2}}{%  \foreach \pos/\lab in {0.5/0, 1.5/1} {%    \path (0, \karnaughmapNumRow-\pos) node[left] {$\lab$};%  }%}

4 rows: anchors are left of the first column with coordinate (0, x) and in the middle of the cells at 0.5 ... 3.5.
\ifthenelse{\equal{\karnaughmapNumRow}{4}}{%  \foreach \pos/\lab in {0.5/00, 1.5/01, 2.5/11, 3.5/10} {%    \path (0, \karnaughmapNumRow-\pos) node[left] {$\lab$};%  }%}

8 rows: anchors are left of the first column with coordinate (0, x) and in the middle of the cells at 0.5 ... 7.5.
\ifthenelse{\equal{\karnaughmapNumRow}{8}}{%  \foreach \pos/\lab in {0.5/000, 1.5/001, 2.5/011, 3.5/010, 4.5/110, 5.5/111, 6.5/101, 7.5/100, 8.5/1100, 9.5/1101, 10.5/1111, 11.5/1110, 12.5/1010, 13.5/1011, 14.5/1001} {%    \path (-0.125, \karnaughmapNumRow-\pos) node[left] {$\lab$};%  }%}

16 rows: anchors are left of the first column with coordinate (0, x) and in the middle of the cells at 0.5 ... 15.5.
\fi

Typeset binary labels above and left of the map if switch \texttt{omitbinaries} is false.

Set base bias of variable labels; the top and left bias is modified depending on if the binary labels are set or not.
\def\karnaughmapVariableBaseBias{0.2}
\ifKV@karnaughmap@omitbinaries
  Set bias to 0 if binaries are not typeset.
  \def\karnaughmapVariableTopBias{0}
  \def\karnaughmapVariableLeftBias{0}
\else
  Set bias accordingly if binaries have 1 or 2 characters.
  \def\karnaughmapVariableTopBias{0.3}
  \ifthenelse{\equal{\karnaughmapVarLabelD}{}}{%    \def\karnaughmapVariableLeftBias{0.2}%  }{%    \def\karnaughmapVariableLeftBias{0.4}%  }
\fi
Typeset the variable labels around the map if switch `omitvariables` is false.

```latex
\ifKV@karnaughmap@omitvariables
\else
\ifthenelse{\equal{\karnaughmapVarLabelA}{}{\%}
  \ifthenelse{\equal{\karnaughmapVarLabelB}{}{\%}
    \draw (1.1, \karnaughmapNumRow + \karnaughmapVariableBaseBias + \karnaughmapVariableTopBias)
    -- node[above] {$\karnaughmapVarLabelA$} ++(0.8, 0);
    \ifKV@karnaughmap@omitnegated
    \else
    \draw (0.1, \karnaughmapNumRow + \karnaughmapVariableBaseBias + \karnaughmapVariableTopBias)
    -- node[above] {$\overline{\karnaughmapVarLabelA}$} ++(0.8, 0);
    \fi
  }{\%}
  \draw (2.1, \karnaughmapNumRow + \karnaughmapVariableBaseBias + \karnaughmapVariableTopBias)
  -- node[above] {$\karnaughmapVarLabelA$} ++(1.8, 0);
  \draw (0.1, \karnaughmapNumRow + \karnaughmapVariableBaseBias)
  -- node[below] {$\overline{\karnaughmapVarLabelA}$} ++(1.8, 0);
  \ifKV@karnaughmap@omitnegated
  \else
  \draw (-\karnaughmapVariableBaseBias -\karnaughmapVariableLeftBias, 0.1)
  -- node[left] {$\karnaughmapVarLabelC$} ++(0, 0.8);
  \fi
 \fi}
\fi
\ifthenelse{\equal{\karnaughmapVarLabelC}{}{\%}
  \ifthenelse{\equal{\karnaughmapVarLabelD}{}{\%}
    \draw (-\karnaughmapVariableBaseBias -\karnaughmapVariableLeftBias, 0.1)
    -- node[left] {$\karnaughmapVarLabelC$} ++(0, 0.8);
    \draw (\karnaughmapNumCol + \karnaughmapVariableBaseBias, 1.1)
    -- node[right] {$\overline{\karnaughmapVarLabelD}$} ++(0, 1.8);
    \ifKV@karnaughmap@omitnegated
    \else
    \draw (-\karnaughmapVariableBaseBias -\karnaughmapVariableLeftBias, 2.1)
    -- node[left] {$\overline{\karnaughmapVarLabelC}$} ++(0, 1.8);
    \draw (\karnaughmapNumCol + \karnaughmapVariableBaseBias, 0.1)
  }{\%}
\fi
\fi
\end{verbatim}

34
Typeset decimal index if switch omitidx is false.
\ifKV@karnaughmap@omitidx
\else
\karnaughmapPrintCellContents[anchor=west]{0.0}{-0.2}{\karnaughmapSize}{\karnaughmapNumRow}{\karnaughmapPrintIndex[\karnaughmapNumVar]}
\fi
\fi

Typeset the cell entries.
\karnaughmapPrintCellContents{0.5}{-0.5}{\karnaughmapSize}{\karnaughmapNumRow}{\karnaughmapPrintValue{\karnaughmapCellEntries}}
\end{scope}
Finishing macro bracket.

\karnaughmapcolorfield
The coloring macro takes two mandatory and one optional argument:
\[\{\text{key}=value\}\{\text{map size}\}\{\text{fields}\}\{\text{color}\}\].
\newcommand{\karnaughmapcolorfield}{[4]}{[%
\setkeys{\karnaughmap}{\text{fill},\text{opacity}}%
\ifthenelse{\equal{#1}{}}{}{\setkeys*[\karnaughmap]{#1}%%
\ifthenelse{\equal{#2}{1}}{\StrSplit{#3}{1}{\karnaughmapColorfieldCur}{\karnaughmapColorfieldRem}}{%
\ifthenelse{\equal{#2}{3}}{\karnaughmapShadeMapfieldTHREE[\text{km}]{\karnaughmapColorfieldCur}{\karnaughmapColorfieldRem}{#4}}}{}%
\ifthenelse{\equal{#2}{2}}{\karnaughmapShadeMapfieldFOUR[\text{km}]{\karnaughmapColorfieldCur}{\karnaughmapColorfieldRem}{#4}}}{}
\ifthenelse{\equal{#2}{3}}{\karnaughmapShadeMapfieldTHREE[\text{km}]{\karnaughmapColorfieldCur}{\karnaughmapColorfieldRem}{#4}}}{}
\ifthenelse{\equal{#2}{4}}{\karnaughmapShadeMapfieldFOUR[\text{km}]{\karnaughmapColorfieldCur}{\karnaughmapColorfieldRem}{#4}}}{}
\karnaughmapcolorfield[#1][#2][\karnaughmapColorfieldRem][#4]}

9 To Do

- Adjust the map label positions as well as the function label to fit the negated variant.
- Increase supported karnaugh map size: 5 and 6 variables should be feasible in a single karnaugh map
• Provide the possibility to permute the karnaugh map

Change History

v1.0
  General: Initial version

v1.0a
  General: Added a to do section

v1.1
  General: Added binary indices

v1.2
  General: Added negated labels and adjusted the positioning of variable and function labels

v1.3
  General: Made the table color adjustable

v1.4
  General: Added a colored box highlight macro

v2.0
  General: Minor touchup for publication

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.

B
\begin 264, 272, 284, 312, 401
\define@cmdkeys 24
\define@key 18, 21, 22
\draw 260, 405, 408, 413, 416, 484, 488, 492, 494, 498, 500, 502, 510, 514, 518, 520, 524, 526, 528
\end 269, 281, 301, 306, 539

C
\cmdKV@karnaughmap@defaultmap 314
\cmdKV@karnaughmap@dontcare 330
\cmdKV@karnaughmap@function 419
\cmdKV@karnaughmap@variables 335, 337, 344–346, 353–356, 363, 365, 373, 375, 383, 385, 393, 395
\cmdKV@karnaughmap@xshift 401
\cmdKV@karnaughmap@yshift 402
\colon 419
\colorlet 19–21, 23

D
\foreach 406, 411, 425, 430, 435, 440, 446, 451, 456, 461
\ifcase 38, 43, 48
\ifdefKV@karnaughmap@binaryidx 36
\ifdefKV@karnaughmap@fill 256
\ifdefKV@karnaughmap@omitbinaries
\ifdefKV@karnaughmap@function
\ifdefKV@karnaughmap@variables

I
\ifKV@karnaughmap@binaryidx 36
\ifdefKV@karnaughmap@fill 256
\ifdefKV@karnaughmap@omitbinaries

36
<table>
<thead>
<tr>
<th>P</th>
<th>\path</th>
<th>64, 73, 90, 107, 136, 165, 219, 257, 420, 421, 426, 431, 436, 442, 447, 452, 457, 463</th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>\pos</td>
<td>425, 426, 430, 431, 435, 436, 440, 442, 446, 447, 451, 452, 456, 457, 461, 463</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>\RequirePackage</td>
<td>1–4</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>S</td>
<td>\scriptsize</td>
<td>35</td>
</tr>
<tr>
<td>\setcounter</td>
<td>7, 8, 332, 333</td>
<td></td>
</tr>
<tr>
<td>\setkarnaughmap</td>
<td>25, 33</td>
<td></td>
</tr>
<tr>
<td>\setkeys</td>
<td>11, 27, 29, 31, 304, 362, 372, 382, 392, 542, 543</td>
<td></td>
</tr>
<tr>
<td>\stepcounter</td>
<td>56, 59</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>\thekarnaughmapIdxCounter</td>
<td>54</td>
</tr>
<tr>
<td>\thekarnaughmapStrCounter</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>\value</td>
<td>38, 43, 48</td>
</tr>
<tr>
<td>X</td>
<td>\x</td>
<td>406–408, 411–413</td>
</tr>
<tr>
<td>\XKV@rm</td>
<td>547, 550, 553</td>
<td></td>
</tr>
</tbody>
</table>