Typesetting Karnaugh Maps and Veitch Charts with LaTeX

Andreas W. Wieland
eMail: `awwieland@gmx.de`

January 7th, 2002

1 Abstract

Karnaugh maps and Veitch charts[1] are used to simplify logic equations. The drawing of them used to be a boring, annoying and error-prone task. With the included macros this is no longer a problem. They can typeset Karnaugh maps and Veitch charts with up to ten variables, which is more than you might likely need. You only have to provide a list of variable identifiers plus the function table of your logic function(s). The macros also allow to mark the simplifications of your logic function directly within a Karnaugh map or Veitch chart.

2 Introduction

Karnaugh maps and Veitch charts are used to simplify logic equations. They are map representations of logic functions, and in that they are equivalent. The difference between the two kinds of diagrams is the way in which the variables of the functions are located within the diagrams: "The Karnaugh map ... displays a function's discriminants according to the reflected binary (Gray) code. The Veitch chart ... displays the discriminants in natural binary order." (F. M. Brown)

However, the drawing of either kind of them used to be a boring, annoying and error-prone task. Since I had to typeset a 50+ page collection of exercises for a basic computer-science lecture with plenty of Karnaugh maps in it, I decided to write some macros that would do this task for me. Let us start first with an introduction on how to use these Macros:

The first thing you have to do is to put the macro file `kvmacros.tex` into a directory where TeX will find it, i.e. you have to put it into a directory where your `TEXINPUTS` environment variable points to. You can then load them by typing `\input kvmacros` somewhere in the preamble of your document.

[1] In previous releases of this document I referred to Karnaugh maps as Karnaugh-Veitch-Maps. This was definitely not correct. Thanks to F. M. Brown for pointing this error out to me.

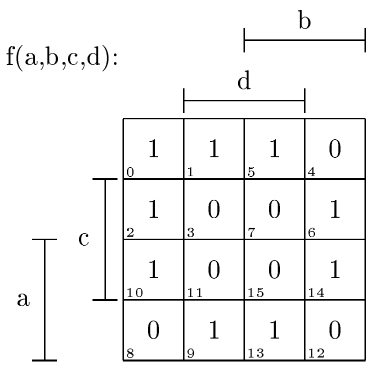
Suppose now you have a logic function f with the following function table:

Index	a	b	c	d	f	Index	a	b	c	d	f
0	0	0	0	0	1	8	1	0	0	0	0
1	0	0	0	1	1	9	1	0	0	1	1
2	0	0	1	0	1	10	1	0	1	0	1
3	0	0	1	1	0	11	1	0	1	1	0
4	0	1	0	0	0	12	1	1	0	0	0
5	0	1	0	1	1	13	1	1	0	1	1
6	0	1	1	0	1	14	1	1	1	0	1
7	0	1	1	1	0	15	1	1	1	1	0

This logic function can easily be put into a Karnaugh map by using the `\karnaughmap`-macro[2]. It has five parameters: The number of variables in the map, an identifier for the function, a list of variable identifiers for the variables, and the list of values of f for each line in the function table. The variable identifiers in the third parameter are ordered from highest to lowest significance (the same way as in the function table, with a having a significance of $2^3 = 8$ and d having a significance of $2^0 = 1$). The list of values of f was read from lowest to highest index. The fifth parameter remains empty in this example, it will be discussed below:

```
\karnaughmap{4}{f(a,b,c,d):}{abcd}{1110011001100110}{}
```

This produces the following Karnaugh map, where the indices in the lower left corner of each box correspond to the indices in the function table:[3]

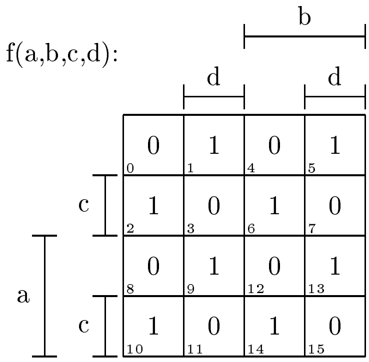


The macros work similar for Veitch charts:

```
\veitchchart{4}{f(a,b,c,d):}{abcd}{0110011001100110}{}
```

[2] Users of older versions of the macro package will notice that the name has changed. Below you will find a discussion of the changes between versions.

[3] The indices can easily be calculated from the variable values in the function table, i.e. line 11: the index equals $a \cdot 2^3 + b \cdot 2^2 + c \cdot 2^1 + d \cdot 2^0 = 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 11$.



Notice the difference in the variable location in the diagrams. In my opinion Karnaugh maps are much easier to use if you need to simplify a logic function; therefore Veitch charts will not be discussed any further in this document. Except otherwise noticed, the macros' syntax and functionality are exactly the same for Karnaugh maps and Veitch charts.

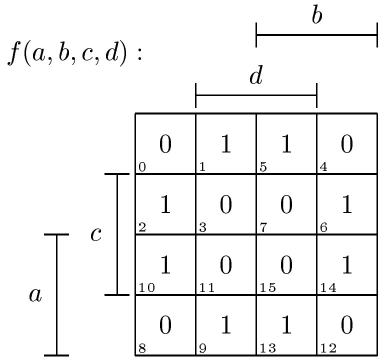
The macros that read the variable list and the list of logic values (i.e. parameters #3 and #4) work recursively. Therefore, you have to take special care to supply the correct number of tokens to print: If you supply a list of variable identifiers or a list of logic values that contains fewer elements than needed the macros will cause error messages similar to the following:

```
! Argument of \kvgetonechar has an extra }.
<inserted text>
                \par
l.201 }{}

?
```

On the contrary, if any of the two lists is longer than needed, *you will not be notified!* Moreover, each entry has to be one character long, otherwise — like a variable identifier enclosed in $s — you have to put it into braces:

```
\karnaughmap{4}{$f(a,b,c,d):$}{{$a$}{$b$}{$c$}{$d$}}{0110011001100110}{}
```

This produces the following Karnaugh map:

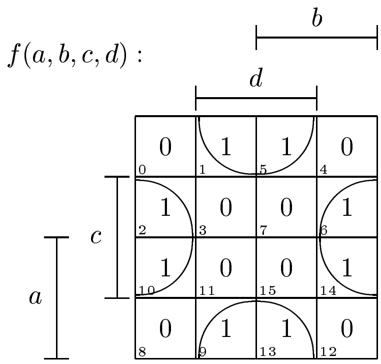


3 Marking simplifications

The already mentioned fifth parameter can be used if you want to draw something inside the Karnaugh map. For example this is useful if you want to show how you simplified a logic function:

```
\karnaughmap{4}{$f(a,b,c,d):$}{{$a$}{$b$}{$c$}{$d$}}%
{0110011001100110}%
{%
\put(0,2){\oval(1.9,1.9)[r]}
\put(4,2){\oval(1.9,1.9)[l]}
\put(2,0){\oval(1.9,1.9)[t]}
\put(2,4){\oval(1.9,1.9)[b]}
}
```

The corresponding Karnaugh map looks like this:



You can use any of LaTeX's graphics macros for this purpose. The Karnaugh map has its pivot point at the lower left point[4] and a unitlength that is

[4]This is not exactly true, but sufficient if you do not want to modify the macros.

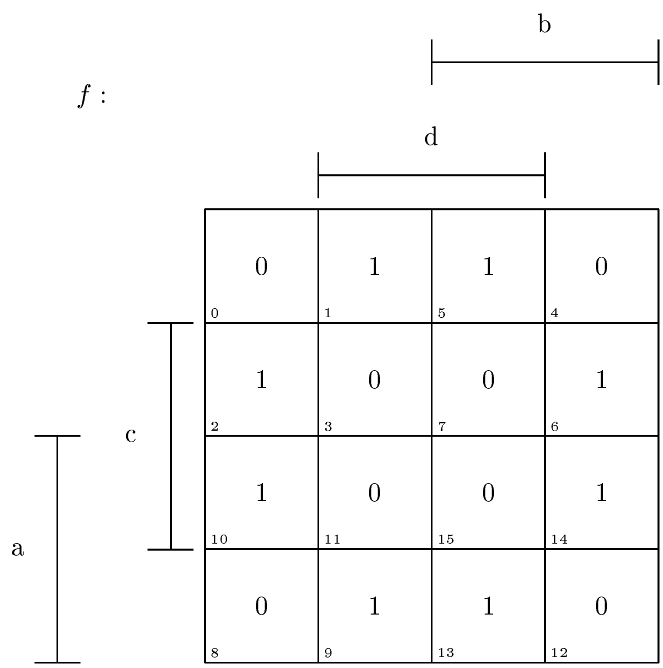
equal to the length of a single box within the Karnaugh map. Sometimes you may find that the `\oval`s that you want to draw inside a Karnaugh map are slightly shifted to the right. This occurs if you use the `eepic`-Package in your document. In this case, make a copy of the `eepic.sty` file, rename it to something like `myeepic.sty` and comment out the definition of the `\oval`-macro in your local copy. If you put this new `myeepic.sty` file somewhere into your TeX's search path so that TeX can find it and you load it properly with e.g. `\usepackage{myeepic}`, you shouldn't experience any more trouble.

4 Other features

There are some more features that you can use. Possibly the most important is that you can change the size of the diagrams by changing the size of the boxes within the map, simply by typing:

```
\kvunitlength=15mm
\karnaughmap{4}{$f:$}{abcd}{0110011001100110}{}
```

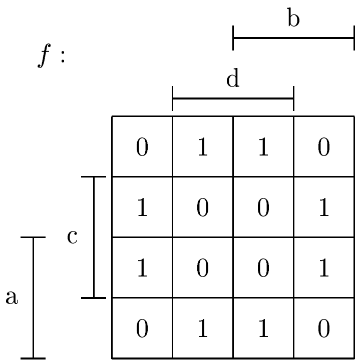
This results in the following Karnaugh map. The setting of the `\kvunitlength` remains active until you change it again;[5] the default `\kvunitlength` is 8 mm:



[5] Or, of course, until you leave the group in which you redefined the value. This applies to all changeable parameters and macro definitions in this section!

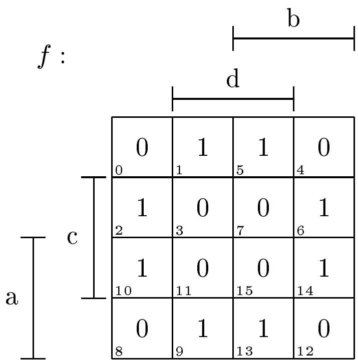
Another feature is that you can switch off the indices inside the map by typing:

```
\kvnoindex
\karnaughmap{4}{$f:$}{abcd}{0110011001100110}{}
```



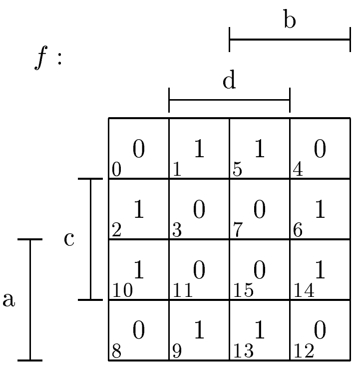
This setting also remains active until you switch indexing on again using the `\kvindex`-macro:

```
\kvindex
\karnaughmap{4}{$f:$}{abcd}{0110011001100110}{}
```



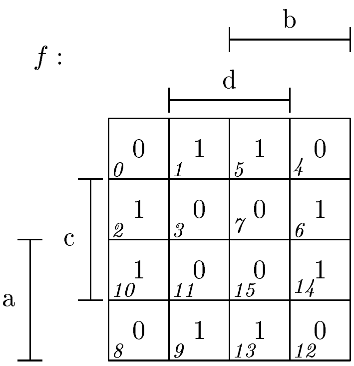
Usually, the font size of the map's contents and indices should be suitable (the defaults are `\tiny` for `\kvindexsize` and `\normalsize` for `\kvcontentsize`). If you want to change them, you can do this by redefining the `\kvcontentsize`- and `\kvindexsize`-macros (Preferably, you should do this in the preamble of your document):

```
\renewcommand\kvindexsize{\footnotesize}
\karnaughmap{4}{$f:$}{abcd}{0110011001100110}{}
```



In the same way you can also change the font family (Although I doubt that this is sensible):

```
\renewcommand\kvindexsize{\footnotesize\it}
\karnaughmap{4}{$f:$}{abcd}{0110011001100110}{}
```



These macros are named `\kvfoo`, because they work both for Karnaugh maps and Veitch charts.

5 If you use an older version of the macros. . .

. . . you will certainly have noticed a number of changes. The most important one is that I have changed the name of the macro that draws Karnaugh maps from `\kvmap` to `\karnaughmap`. I suggest that you use the new name in documents that you newly create. However, I introduced an alias so that the old macro name still works:

```
\def\kvmap{\karnaughmap}
```

Also, in some older versions tokens in the variable list or function table that were longer than one character had to be enclosed in *double* braces (If you have used any of these versions you will certainly remember this annoying feature). This is no longer necessary — however, it does no harm.

If you need to know more about the exact changes, please consult the revision history in `kvmacros.tex`.

6 Final remarks

This is all you need to know about the usage of my macros. I have tried them out and I haven't yet found any serious bugs, but I'm pretty sure there are plenty. In this case, or if you have comments or suggestions, please send me an eMail. Anyway, I would very much appreciate if you would notify me when you use my macros; I would like to know if they are of any use to somebody.

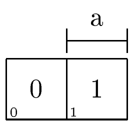
The maximum size map I could produce was a Karnaugh map with 10 variables; with bigger maps I only exceeded TEX's main memory. This is due to the macros' recursive algorithm. Quite likely you will exceed TEX's capacity with even smaller maps if they occur in large documents, but I think no one in his right mind will ever use Karnaugh maps of more than 6 variables, so I didn't care about this problem.

7 More examples

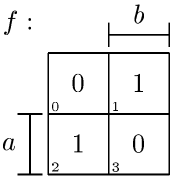
Below I have given some more examples of Karnaugh maps produced with my macros:

```
\karnaughmap{1}{}{a}{01}{}

\karnaughmap{2}{$f:$}{{$a$}{$b$}}{0110}{}

\karnaughmap{3}{$f:$}{{$a$}{$b$}{$c$}}{01100110}{}

\karnaughmap{4}{$f:$}{{$a$}{$b$}{$c$}{$d$}}{0110011001100110}{}

\karnaughmap{4}{$f:$}{abcd}{abcdefghijklmnop}{}

\karnaughmap{5}{$f:$}{{$a$}{$b$}{$c$}{$d$}{$e$}}%
{0123456789abcdef0123456789abcdef}{}

\karnaughmap{6}{$f_i^t:$}{abcdef}%
{0123456789{10}{11}{12}{13}{14}{15}%
{16}{17}{18}{19}{20}{21}{22}{23}%
{24}{25}{26}{27}{28}{29}{30}{31}%
{32}{33}{34}{35}{36}{37}{38}{39}%
{40}{41}{42}{43}{44}{45}{46}{47}%
{48}{49}{50}{51}{52}{53}{54}{55}%
{56}{57}{58}{59}{60}{61}{62}{63}}{}%
```

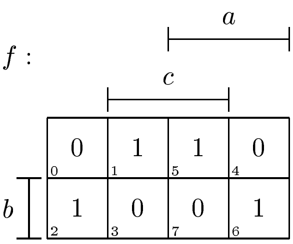
```
\kvunitlength=6mm
\kvnoindex
\karnaughmap{8}{$q_2^4:$}{{$a$}{$b$}{$c$}{$d$}{$e$}%
{$f$}{$g$}{$h$}}{%
0011001111001100%
0011001111001100%
0011001111001100%
0011001111001100%
0011001111001100%
0011001111001100%
0011001111001100%
0011001111001100%
0011001111001100%
0011001111001100%
0011001111001100%
0011001111001100%
0011001111001100%
0011001111001100%
0011001111001100%
0011001111001100%
}{}
\kvunitlength=8mm
\kvindex
```



f :



f :



f :

		d	d, b	b
	0_{0}	1_{1}	1_{5}	0_{4}
c	1_{2}	0_{3}	0_{7}	1_{6}
c, a	1_{10}	0_{11}	0_{15}	1_{14}
a	0_{8}	1_{9}	1_{13}	0_{12}

f :

		d	d, b	b
	a_{0}	b_{1}	f_{5}	e_{4}
c	c_{2}	d_{3}	h_{7}	g_{6}
c, a	k_{10}	l_{11}	p_{15}	o_{14}
a	i_{8}	j_{9}	n_{13}	m_{12}

f :

		e	e, c	c	c, a	c, e, a	e, a	a
	0_{0}	1_{1}	5_{5}	4_{4}	4_{20}	5_{21}	1_{17}	0_{16}
d	2_{2}	3_{3}	7_{7}	6_{6}	6_{22}	7_{23}	3_{19}	2_{18}
d, b	a_{10}	b_{11}	f_{15}	e_{14}	e_{30}	f_{31}	b_{27}	a_{26}
b	8_{8}	9_{9}	d_{13}	c_{12}	c_{28}	d_{29}	9_{25}	8_{24}

f_i^t :

b

d

f f

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

q_2^4 :

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8 Sourcecode

Here now you will find the current sourcecode of my macros. I should have commented it more extensively, but failed due to lack of time. Sorry.

```
%% This is file `kvmacros.tex',
%% Version of January 7th, 2002
%%
%% Copyright (C) 1998-2002 by Andreas W. Wieland, awwieland@gmx.de
%%
%% This program can be redistributed and/or modified under the terms
%% of the LaTeX Project Public License distributed from CTAN
%% archives in directory macros/latex/base/lppl.txt; either
%% version 1 of the License, or (at your option) any later version,
%% with `The Program' referring to the software `kvmacros.tex' and its
%% accompanying documentation and `The Copyright Holder' referring to the
%% person Andreas W. Wieland.
```

```
%%
%%
%%
%% IMPORTANT NOTICE:
%%
%% For error reports, comments or suggestions in case of UNCHANGED
%% versions send mail to:
%% awwieland@gmx.de
%%
%% Anyway, if you use my macros, send me an eMail, too. I would like to know
%% if they are useful to somebody out there.
%%
%% Please do not request updates from me directly. Distribution is
%% done through Mail-Servers and TeX organizations.
%%
%% You are allowed to distribute this file under the condition that
%% it is distributed together with `kvdoc.tex', `kvdoc.dvi' and `kvdoc.ps'.
%%
%% If you receive this file alone from someone, complain!
%%
\typeout{}
\typeout{Macros for typesetting Karnaugh maps and Veitch charts}
\typeout{Version of January 7th, 2002}
\typeout{by Andreas W. Wieland, awwieland@gmx.de}
\typeout{}
%%
%% Change History:
%% August 04th, 1998: Original Version
%%
%% August 19th, 1998: Minor changes to make the lines within the maps look
%% better, also moved the variable identifiers from the right to the left and
%% added an identifier for the logic function; since then \kvmap has 5
%% instead of 4 parameters. I also included the documentation in an additional
%% Postscript-file.
%%
%% September 23rd, 1998: Fixed the problem with double braces by introducing
%% the \compdummy and \ende macros in the \ifx-comparisons (see below):
%% The parameters that are longer than one character do not have to be
%% enclosed in double braces any longer --- single braces are
%% sufficient. However, double braces will still be processed correctly.
%%
%% May 19th, 1999: Changed the documentation (which was months overdue) after
%% F. M. Brown pointed me to the incorrectness of the term
%% "Karnaugh-Veitch-map". Actually, the macros itself haven't
%% changed. Probably I should introduce macros for typesetting Veitch charts
%% as well (which would be a minor task). This version was actually never
%% released.
%%
%% May 25th, 1999: Introduced macros for Veitch charts
%% (\veitchfoo) and renamed the macros that are solely related to
```

```
%% Karnaugh maps from \kvfoo to \karnaughfoo. Introduced aliases for those
%% macros that are user-callable to maintain compatibility with older versions:
%% \kvmap -> \karnaughmap. However, \kvindex, \kvnoindex and \kvunitlength
%% haven't changed their names as they are related to both Karnaugh maps and
%% Veitch charts. Changed the (plain stupid) algorithm that processes the
%% variable-length arguments so that it takes much less time to produce large
%% maps. Several macros so became redundant and are no longer included. Also I
%% introduced \kvindexsize and \kvcontentsize to adjust the font size of the
%% indices and the contents of a diagram. They default to \tiny and
%% \normalsize.
%%
%% January 17th, 2001: Changed the eMail-address in the macro file and the
%% accompanying documentation and made minor changes to the documentation.
%%
%% January 7th, 2002: Exchanged the \xdef to \gdef in the definition of
%% \kvargumentstring and \kvgetonechar so that things like {\textbf 1} in the
%% argumentlist of \karnaughmap do not cause error messages any longer.
%% Thanks to Michal Medveck{\'y} for pointing me to this error.
%%
%%
%%
%% We need a fixed dimension for a single field in a Karnaugh map:
%%
\newdimen\kvunitlength
\kvunitlength=8mm
%%
%% We need a default font size for the indices:
%%
\newcommand{\kvindexsize}{\tiny}
%%
%% And we need a default size for the contents and variable identifiers:
%%
\newcommand{\kvcontentsize}{\normalsize}
%%
%% First, we have to introduce some counters:
%%
%% \kvrecursiondepth is used to control the recursion of the
%% \karnaughmakemap- and \veitchmakechart-macro.
%%
\newcount\kvrecursiondepth
%%
%% The \kvindexcounter is needed for the indices in the fields of the
%% diagrams.
%%
\newcount\kvindexcounter
%%
%% \kvxsize and \kvysize store the dimensions of an entire diagram.
%%
\newcount\kvxsize
\newcount\kvysize
```

```
%%
%% Some counters are necessary to compute the marks for the variable
%% identifiers:
%%
\newcount\kvvarno
\newcount\kvxvarno
\newcount\kvyvarno
\newcount\kvmarkstart
\newcount\kvmarklength
\newcount\kvmarknum
\newcount\kvmarkmove
%%
%% And we need a savebox to store the variable marks:
%%
\newsavebox\kvsavebox
%%
%% Single fields in a diagram should be indexed, which makes the map easier to
%% use. This is the default. If you don't want indices, simply call
%% \kvnoindex. If you want indices back, call \kvindex:
%%
\def\kvnoindex{%
\def\kvcurrentindex{}}
%%
\def\kvindex{%
\def\kvcurrentindex{%
\the\kvindexcounter\global\advance\kvindexcounter by 1}%
}
%%
\kvindex
%%
%% We need a macro that computes the powers of two:
%%
\def\kvpoweroftwo#1#2{% Computes #1=2^#2, both of which have to be counters
{\ifnum#2>0
\global\multiply#1 by 2
\advance#2 by -1
\kvpoweroftwo{#1}{#2}
\fi}}
%%
%% The macros \kvargumentstring, \kvgetchar and \kvgetonechar are needed to
%% process the variabe-length parameters in \karnaughmap and \veitchchart:
%%
\def\kvargumentstring#1{\gdef\kvdummystring{#1{}\noexpand\end}}
%%
\def\kvgetchar{\expandafter\kvgetonechar\kvdummystring}
%%
\def\kvgetonechar#1#2\end{{#1}\gdef\kvdummystring{#2\noexpand\end}}%
%%
%% The Macro \karnaughmakemap calls itself recursively until the parameter #1
%% equals 1, whereupon it returns a single token from the list of arguments,
```

```
%% enclosed in a \makebox plus the index (if enabled) in a smaller \makebox:
%%
\def\karnaughmakemap#1#2{{%
\kvrecursiondepth=\number#1
\ifnum\kvrecursiondepth>1
\divide\kvrecursiondepth by 2
\unitlength=\kvunitlength
\multiply\unitlength by \kvrecursiondepth
%%
\ifcase#2
%%
%% The parameter #2 of \karnaughmakemap is needed because the inner Karnaugh
%% maps need to be mirrored. This is achieved by the following case-statement,
%% which orders the inner Karnaugh maps properly:
%%
%% Case 0: top-left Karnaugh map
\begin{picture}(2,2)%
\put(0,1){\karnaughmakemap{\kvrecursiondepth}{0}}%
\put(1,1){\karnaughmakemap{\kvrecursiondepth}{1}}%
\put(0,0){\karnaughmakemap{\kvrecursiondepth}{2}}%
\put(1,0){\karnaughmakemap{\kvrecursiondepth}{3}}%
\end{picture}%
\or
%% Case 1: top-right Karnaugh map
\begin{picture}(2,2)%
\put(1,1){\karnaughmakemap{\kvrecursiondepth}{1}}%
\put(0,1){\karnaughmakemap{\kvrecursiondepth}{0}}%
\put(1,0){\karnaughmakemap{\kvrecursiondepth}{3}}%
\put(0,0){\karnaughmakemap{\kvrecursiondepth}{2}}%
\end{picture}%
\or
%% Case 2: bottom-left Karnaugh map
\begin{picture}(2,2)%
\put(0,0){\karnaughmakemap{\kvrecursiondepth}{2}}%
\put(1,0){\karnaughmakemap{\kvrecursiondepth}{3}}%
\put(0,1){\karnaughmakemap{\kvrecursiondepth}{0}}%
\put(1,1){\karnaughmakemap{\kvrecursiondepth}{1}}%
\end{picture}%
\or
%% Case 3: bottom-right Karnaugh map
\begin{picture}(2,2)%
\put(1,0){\karnaughmakemap{\kvrecursiondepth}{3}}%
\put(0,0){\karnaughmakemap{\kvrecursiondepth}{2}}%
\put(1,1){\karnaughmakemap{\kvrecursiondepth}{1}}%
\put(0,1){\karnaughmakemap{\kvrecursiondepth}{0}}%
\end{picture}%
\fi
\else
\unitlength=\kvunitlength
\begin{picture}(1,1)
```

```
\put(0,0){\makebox(1,1){\kvcontentsize\kvgetchar}}%
\put(0.05,0.05){\makebox(0.9,0.9)[bl]{\kvindexsize\kvcurrentindex}}%
\end{picture}
\fi}}%
%%
%% \karnaughmaketopmark typesets the variable marks of a Karnaugh map that are
%% located on top of the diagram:
%%
\def\karnaughmaketopmark{%
  \unitlength\kvunitlength
  \begin{picture}(\kvxsize,1)
    \kvmarkstart=1
    \kvpoweroftwo{\kvmarkstart}{\kvxvarno} % \kvymarkstart is the start
                                % position for the \multiput
    \kvmarklength=\kvmarkstart
    \multiply\kvmarklength by 2 % \kvmarklength is the length of a mark
    \kvmarkmove=\kvmarkstart
    \multiply\kvmarkmove by 4 % This is the move distance for the \multiput.
    \kvmarknum=\kvxsize
    \divide\kvmarknum by \kvmarkmove % This is the number of repetitions for
                                % the \multiput.
    %The highest-order variable mark needs a special treatment:
    \ifnum\kvmarknum=0\kvmarknum=1\divide\kvmarklength by 2\fi
    \savebox\kvsavebox(\kvmarklength,1){%
      \begin{picture}(\kvmarklength,1)
      \put(0,0.3){\makebox(\kvmarklength,0.7){\kvcontentsize\kvgetchar}}
      \put(0,0.1){\line(0,1){0.4}}
      \put(\kvmarklength,0.1){\line(0,1){0.4}}
      \put(0,0.3){\line(1,0){\kvmarklength}}
      \end{picture}}
    \multiput(\kvmarkstart,0)(\kvmarkmove,0){\kvmarknum}{\usebox\kvsavebox}
  \end{picture}
}
%%
%% \karnaughmakeleftmark typesets the variable marks of a Karnaugh map that are
%% located on the left of the diagram:
%%
\def\karnaughmakeleftmark{%
  \unitlength\kvunitlength
  \begin{picture}(-1,\kvysize)(0,-\kvysize)
    \kvmarkstart=1
    \kvpoweroftwo{\kvmarkstart}{\kvyvarno} % \kvmarkstart is the start
                                % position for the \multiput
    \kvmarklength=\kvmarkstart
    \multiply\kvmarklength by 2 % \kvmarklength is the length of a mark
    \kvmarkmove=\kvmarkstart
    \multiply\kvmarkmove by 4 % This now is the move distance for the
                              % \multiput.
    \kvmarknum=\kvysize
    \divide\kvmarknum by \kvmarkmove % This now is the number of
```

```
                              % repetitions for the \multiput.
    %The highest-order variable mark needs a special treatment:
    \ifnum\kvmarknum=0\kvmarknum=1\divide\kvmarklength by 2\fi
    \advance\kvmarkstart by \kvmarklength
    \savebox\kvsavebox(1,\kvmarklength){%
      \begin{picture}(1,\kvmarklength)
        \put(-0.3,0){\makebox(-0.7,\kvmarklength){\kvcontentsize\kvgetchar}}
        \put(-0.1,0){\line(-1,0){0.4}}
        \put(-0.1,\kvmarklength){\line(-1,0){0.4}}
        \put(-0.3,0){\line(0,1){\kvmarklength}}
      \end{picture}}
    \multiput(0,-\kvmarkstart)(0,-\kvmarkmove){\kvmarknum}{\usebox\kvsavebox}
  \end{picture}
}
%% \karnaughmakemarks calls \karnaughmaketopmark or \karnaughmakeleftmark
%% depending on whether \kvvarno is odd or even.
%%
\def\karnaughmakemarks{%
\ifnum\kvvarno>0
  \let\next=\karnaughmakemarks
  \ifodd\kvvarno % We have to make a mark at the top
    \advance\kvxvarno by -1
    \put(0,\kvxvarno){\karnaughmaketopmark}
  \else % We have to make a mark at the left
    \advance\kvyvarno by -1
    \put(-\kvyvarno,-\kvysize){\karnaughmakeleftmark}
  \fi
  \advance\kvvarno by -1
\else
  \let\next=\relax
\fi
\next
}
%%
%% \karnaughmap is the macro that a user calls if he wants to draw a
%% Karnaugh map:
%%
\def\karnaughmap#1#2#3#4#5{%
%%
%% #1 is the number of variables in the Karnaugh map
%% #2 is the identifier of the function
%% #3 is the list of identifiers of those variables
%% #4 is the list of tokens that have to be written into the map
%% #5 is something that you want to draw inside the Karnaugh map
%%
\kvvarno=#1  % \kvvarno is the total number of variables
\kvyvarno=#1 % \kvyvarno is the number of variable marks at the left
\divide\kvyvarno by 2
\kvxvarno=#1 % \kvxvarno is the number of variable marks on top
\advance\kvxvarno by -\kvyvarno
```

```
\kvxsize=1
\kvpoweroftwo{\kvxsize}{\kvxvarno}
\kvysize=1
\kvpoweroftwo{\kvysize}{\kvyvarno}
\advance\kvxsize by \kvyvarno
\advance\kvxvarno by \kvysize
\unitlength\kvunitlength
\begin{picture}(\kvxsize,\kvxvarno)(-\kvyvarno,-\kvysize)
\advance\kvxsize by -\kvyvarno
\advance\kvxvarno by -\kvysize
\put(0,-\kvysize){%
\begin{picture}(\kvxsize,\kvysize)
\multiput(0,0)(0,1){\kvysize}{\line(1,0){\kvxsize}}
\multiput(0,0)(1,0){\kvxsize}{\line(0,1){\kvysize}}
\put(0,\kvysize){\line(1,0){\kvxsize}}
\put(\kvxsize,0){\line(0,1){\kvysize}}
#5
\end{picture}}
\put(-\kvyvarno,0){\makebox(\kvyvarno,\kvxvarno){#2}}
\kvindexcounter=0
\kvargumentstring{#4}
\put(0,-\kvysize){\karnaughmakemap{\kvysize}{0}}
\ifodd\kvvarno
{\divide\kvxsize by 2
\put(\kvxsize,-\kvysize){\karnaughmakemap{\kvysize}{1}}}
\fi
\kvargumentstring{#3}
\karnaughmakemarks
%%
\end{picture}
}%
%%
%% The definition of \kvmap is necessary to maintain compatibility with older
%% versions of this macro package:
%%
\def\kvmap{\karnaughmap}%
%%
%%
%% The Macro \veitchmakechart calls itself recursively until the parameter #1
%% equals 1, whereupon it returns a single token from the list of arguments,
%% enclosed in a \makebox plus the index (if enabled) in a smaller \makebox:
%%
\def\veitchmakechart#1{{%
\kvrecursiondepth=\number#1
\ifnum\kvrecursiondepth>1
\divide\kvrecursiondepth by 2
\unitlength=\kvunitlength
\multiply\unitlength by \kvrecursiondepth
%%
\begin{picture}(2,2)%
```

```
\put(0,1){\veitchmakechart{\kvrecursiondepth}}%
\put(1,1){\veitchmakechart{\kvrecursiondepth}}%
\put(0,0){\veitchmakechart{\kvrecursiondepth}}%
\put(1,0){\veitchmakechart{\kvrecursiondepth}}%
\end{picture}%
\else
\unitlength=\kvunitlength
\begin{picture}(1,1)
\put(0,0){\makebox(1,1){\kvcontentsize\kvgetchar}}%
\put(0.05,0.05){\makebox(0.9,0.9)[bl]{\kvindexsize\kvcurrentindex}}%
\end{picture}
\fi}}%
%%
\def\veitchmaketopmark{%
  \unitlength\kvunitlength
  \begin{picture}(\kvxsize,1)
    \kvmarkstart=1
    \kvpoweroftwo{\kvmarkstart}{\kvxvarno} % \kvymarkstart is the start
                                % position for the \multiput
    \kvmarklength=\kvmarkstart % \kvmarklength is the length of a mark
    \kvmarkmove=\kvmarkstart
    \multiply\kvmarkmove by 2 % This is the move distance for the \multiput.
    \kvmarknum=\kvxsize
    \divide\kvmarknum by \kvmarkmove % This is the number of repetitions for
                                     % the \multiput.
    \savebox\kvsavebox(\kvmarklength,1){%
      \begin{picture}(\kvmarklength,1)
      \put(0,0.3){\makebox(\kvmarklength,0.7){\kvcontentsize\kvgetchar}}
      \put(0,0.1){\line(0,1){0.4}}
      \put(\kvmarklength,0.1){\line(0,1){0.4}}
      \put(0,0.3){\line(1,0){\kvmarklength}}
      \end{picture}}
    \multiput(\kvmarkstart,0)(\kvmarkmove,0){\kvmarknum}{\usebox\kvsavebox}
  \end{picture}
}
%%
\def\veitchmakeleftmark{%
  \unitlength\kvunitlength
  \begin{picture}(-1,\kvysize)(0,-\kvysize)
    \kvmarkstart=1
    \kvpoweroftwo{\kvmarkstart}{\kvyvarno} % \kvmarkstart is the start
                                % position for the \multiput
    \kvmarklength=\kvmarkstart
    \kvmarkmove=\kvmarkstart
    \multiply\kvmarkmove by 2 % This now is the move distance for the \multiput.
    \kvmarknum=\kvysize
    \divide\kvmarknum by \kvmarkmove % This now is the number of
                                     % repetitions for the \multiput.
    \advance\kvmarkstart by \kvmarklength
    \savebox\kvsavebox(1,\kvmarklength){%
```

```
      \begin{picture}(1,\kvmarklength)
        \put(-0.3,0){\makebox(-0.7,\kvmarklength){\kvcontentsize\kvgetchar}}
        \put(-0.1,0){\line(-1,0){0.4}}
        \put(-0.1,\kvmarklength){\line(-1,0){0.4}}
        \put(-0.3,0){\line(0,1){\kvmarklength}}
      \end{picture}}
    \multiput(0,-\kvmarkstart)(0,-\kvmarkmove){\kvmarknum}{\usebox\kvsavebox}
  \end{picture}
}
%%
\def\veitchmakemarks{%
\ifnum\kvvarno>0
  \let\next=\veitchmakemarks
  \ifodd\kvvarno % We have to make a mark at the top
    \advance\kvxvarno by -1
    \put(0,\kvxvarno){\veitchmaketopmark}
  \else % We have to make a mark at the left
    \advance\kvyvarno by -1
    \put(-\kvyvarno,-\kvysize){\veitchmakeleftmark}
  \fi
  \advance\kvvarno by -1
\else
  \let\next=\relax
\fi
\next
}
%%
\def\veitchchart#1#2#3#4#5{%
%%
%% #1 is the number of variables in the Veitch chart
%% #2 is the identifier of the function
%% #3 is the list of identifiers of those variables
%% #4 is the list of tokens that have to be written into the chart
%% #5 is something that you want to draw inside the Veitch chart
%%
\kvvarno=#1  % \kvvarno is the total number of variables
\kvyvarno=#1 % \kvyvarno is the number of variable marks at the left
\divide\kvyvarno by 2
\kvxvarno=#1 % \kvxvarno is the number of variable marks on top
\advance\kvxvarno by -\kvyvarno
\kvxsize=1
\kvpoweroftwo{\kvxsize}{\kvxvarno}
\kvysize=1
\kvpoweroftwo{\kvysize}{\kvyvarno}
\advance\kvxsize by \kvyvarno
\advance\kvxvarno by \kvysize
\unitlength\kvunitlength
\begin{picture}(\kvxsize,\kvxvarno)(-\kvyvarno,-\kvysize)
\advance\kvxsize by -\kvyvarno
\advance\kvxvarno by -\kvysize
```

```
\put(0,-\kvysize){%
\begin{picture}(\kvxsize,\kvysize)
\multiput(0,0)(0,1){\kvysize}{\line(1,0){\kvxsize}}
\multiput(0,0)(1,0){\kvxsize}{\line(0,1){\kvysize}}
\put(0,\kvysize){\line(1,0){\kvxsize}}
\put(\kvxsize,0){\line(0,1){\kvysize}}
#5
\end{picture}}
\put(-\kvyvarno,0){\makebox(\kvyvarno,\kvxvarno){#2}}
\kvindexcounter=0
\kvargumentstring{#4}
\put(0,-\kvysize){\veitchmakechart{\kvysize}}
\ifodd\kvvarno
{\divide\kvxsize by 2
\put(\kvxsize,-\kvysize){\veitchmakechart{\kvysize}}}
\fi
\kvargumentstring{#3}
\veitchmakemarks
%%
\end{picture}
}%
%%

%%% Local Variables:
%%% mode: latex
%%% TeX-master: t
%%% End:
```