

Typesetting Crosswords via T_EX

Kees van der Laan

March 1992

Abstract

A macro is provided for typesetting crosswords via (plain) T_EX, or any T_EX, which allows `\halign` use. The specification of the crossword information can be done in the WYSIWYG way,¹ and does not require `\halign` markup, just the data.

Keywords

Crosswords, games, plain T_EX.

1 Introduction

This work emerged from my work on tables and FIFO—First In First Out.²

Hamilton Kelley (1990) published sophisticated L^AT_EX macros for ‘drawing’ crosswords.

Although not a crossworder myself, it comes to mind that typesetting crosswords comes down to

- specification of the puzzle and the solution
- providing the clues
- typesetting it all.

It is not difficult to use `\halign`. The difficulty is to keep it simple and flexible, to adopt simple conventions, and to allow natural input.

2 Examples

First the puzzle, next the clues and finally the solution to the puzzle are given as examples of use.

2.1 Puzzle

1	■	2	
3	4		■
■		■	5
6			

is obtained via

```
\input crw.tex
```

¹What You See Is What You Get

²Table Diversions, to be presented at EuroT_EX92, and FIFO and LIFO incognito, submitted for publication.

³A ‘white lie,’ spaces are partly allowed for crossed out cells, see the Appendix.

⁴Admitted, it is not trivial to have the clues formatted with the numbers hanging out without mark up information in the text. This can be handled by a macro with two arguments and a space as separator to distinguish the clue number from the clue text. This macro can be invoked after lines have been taken apart via `\bfifol... \efifol`. It is analogous to the nesting applied in `\markup`. This can be suitably applied within a `\valign`.

```
\bdata%
```

```
P*On
```

```
DEk*
```

```
*n*S
```

```
Edit
```

```
\edata\markup\data
```

```
$$\puzzlettrue\crw\data$$
```

Conventions for `\bdata {data} \edata3`

- cell descriptions have to be given per line
- * denotes crossed out cell
- capitals denote marked open cells (with reference numbers to the clues), and letters of the solution
- lower case letters, denote empty cells, and letters of the solution.

I chose to provide the crossword information via the above given conventions, because it is natural and can be prepared independently. I refrained from the interactive provision of the information within T_EX, because I don’t know how to correct typos easily in that way.

`\markup\data` inserts the mark up necessary for the expansion of `\crw`, via redefinition of `\data`.

`\puzzlettrue\crw\data` typesets the puzzle.

2.2 Clues

Is there a problem? Just columns of text, eventually aligned.⁴ This subsection has nothing to do with the file `crw.tex`.

Across	Down
2 Switch mode	1 Public domain
3 Knuth	2 All right
6 Prior to T _E X	4 All comes to it
	5 Atari type

is obtained via

```
\smallskip\noindent
\vtop{\hsize=21ex\obeylines Across
2 Switch mode
3 Knuth
6 Prior to \TeX
}\vtop{\hsize=25ex\obeylines Down
1 Public domain
2 All right
4 All comes to it
5 Atari type}
```

2.3 Solution

\puzzlefalsetoggles typesetting the solution, in uppercase. After \input crw.tex, data definition and markup, the solution

P		O	N
D	E	K	
	N		S
E	D	I	T

is obtained via

```
$$\def\num{}\crw\data$$
```

Redefinition of \num—\def\num{ }—suppresses the typesetting of the reference numbers for the clues.

3 Programming

Crossword diagrams can be characterized by a table with 1-character elements: (marked) empty cells, crossed out cells, and letters. A cell is \csize by \csize, with height .8\csize. The carriage return, ^^M, is an active character allowing WYSIWYG input. The consequence is that no &-s nor \cr-s are needed in the data.⁵ The nice side-effect is that visual verification of the input is alleviated. The WYSIWYG provided data is transformed into marked up data ready for use within \halign⁶ by the macro \markup. An inefficiency is that the *programming* of the markup is done once, while the mark up will be *inserted* whenever \data is invoked. The macro \crw typesets the crossword. The numbering of the marked cells is done automatically, row-wise and hidden.

3.1 File crw.tex

```
\let\ea=\expandafter \newif\ifpuzzle
\newcount\cnt\newdimen\csize\csize=3ex
%
\def\crw#1{\toxit{\offinterlineskip%
```

```
\halign{\bcell width0pt%of strut
\prc##\ecell&&\bcell\relax
\prc##\ecell\cr#1}%end \halign
\global\cnt0}}%end \crw
%
\def\toxit#1{\vtop{\hrule
\hbox{\vrule\vbox{#1}\vrule}
\hrule}}%end \toxit (\vtop used)
%
\def\bcell{\hbox to\csize\group\vrule
height.8\csize depth.2\csize}
\let\ecell=}%TB 385, \egroup
%
\def\prc#1{\if*#1\cc\else\ifx\relax#1%
\cc\else%no crossed out cell
\ifnum'#1=\lccode'#1\low{#1}\else%
\cap{#1}\fi\fi\fi}%end \prc
\def\cc{\leaders\hrule height.8\csize
depth.2\csize\hfill}%end Crossed Cell
\def\low#1{\ifpuzzle\null\else\hfil%
\uppercase{#1}\fi\hfil}%end \low
\def\cap#1{\num\ifpuzzle\null%
\else\hfil#1\fi\hfil}%end \cap
%
\def\num{\global\advance\cnt1\relax%
\vbox to.8\csize{\rlap{\kern1pt%
\fi\verm\the\cnt\hss}\vfil}}%end \num
%
%furthermore, in order to omit & and \cr
%in the data (natural input)
\def\markup#1{\ea\gdef\ea#1\ea{\ea%
\bfifol#1\efifol}}%end \markup
%
{\catcode'\^^M=13 %local scope
%Pick up and processing of lines
\gdef\bfifol#1^^M#2\efifol{%
\process{#1}%process line
\ifx\empty#2\empty\let\aux1=\lr%
\else\def\aux1{\rs\bfifol#2\efifol}%
\fi\aux1} %end local scope
%
\def\process#1{\bfifo#1\efifo}
%
%Pick up etc of chars per line
\def\bfifo#1#2\efifo{#1%put #1 back
\ifx\empty#2\empty\let\aux=\relax%
\else\def\aux{\cs\bfifo#2\efifo}%
\fi\aux}%end \bfifo... \efifo
%
\def\cs{&}\def\rs{\cr\noalign{\hrule}}
\def\lr{\cr}
%
%next is necessary to allow spaces
%for * (except in last column)
\def\bdata{\bgroup\obeylines%
\obeyspaces\store}
\def\store#1\data{\egroup\def\data
{#1}}
{\obeyspaces\global\let =\relax}
\endinput %27/3/92 cgl@rug.nl
```

⁵When they are provided along with the data, the invoke \markup\data is superfluous.

⁶Via nested use of \bfifo... \efifo, see FIFO and LIFO incognito

`\crw`

The crossword is implemented as a ruled `\halign`, with the framing added separately via `\toxit`. `\offinterlineskip` suppresses the space between the rows of the table. The explicit number of cells is not necessary to specify. The counter for the reference numbers is reset at the end.

`\bcell... \ecell`

The cell size is prescribed via the size of `\hbox` and `\vrule`. For the first column `\vrule` is set to `width0pt`, for the others the default width is used.

`\prc`

The macro typesets the cell contents according to the `\data`. (`\halign` takes care of the (inserted) `&-s`, `\cr-s`, and `\noalign-s`.) `\if*#1` tests whether a crossed out cell has to be typeset, and if so `\cc` is invoked. For the other situation according to the case of the letter `\low(er case letter)` or `\cap(ital letter)`, is invoked.

`\cc`

Crossed out cells are typeset as black cells via application of the leaders mechanism.⁷

`\low, \cap`

How the cell contents will be typeset depends upon the switch `\ifpuzzle`. The letters are typeset in upper case and centered. Note the invoke of `\num`.

`\num`

Generates and typesets the reference numbers in the left upper corners.

`\markup`

The `\data` is processed per line via `\bfifol... \efifol`. The separator `^M` splits the data. The first line is delivered in the first argument and the rest in the second argument. This splitting up is repeated recursively.

`\bfifol... \efifol`

`\bfifo... \efifo` is invoked and `\rs`, respectively `\lr`, is added. `\rs` denotes row separator, and `\lr` stands for last row.

`\bfifo... \efifo`

Inserts column separators, `\cs`.

`\bdata, \store`

These store the user provided information between `\bdata` and `\edata` in `\data`, with the carriage return as active character.

3.2 Some pitfalls

In the macro `\prc` a test is needed to reveal the case of a letter. The PASCAL-like test `\if\lowercase{#1}#1...8` is a pitfall; `\lowercase{\if#1}#1` and `\ifnum`#1>96` work.

The functionality `\bdata {data} \edata` stores the user supplied information with active carriage returns. However,

```
\def\bdata{\bgroup\obeylines\gdef\data
\bgroup}
\def\edata{\egroup\egroup}
```

is a pitfall (the `\bgroup` after `\gdef` must be an *explicit* brace!). In the examples the switch settings and the redefinition of `\num` and `\rs`, have been done *within* the math display, in order to keep the modifications local.

4 It is all in the game

Puzzle, clues and solution can be typeset all at once.

	Across	Down	
	2 Switch mode	1 Public domain	
	3 Knuth	2 All right	
	6 Prior to T _E X	4 All comes to it	
		5 Atari type	

is obtained—after `\input crw.tex`, data definition and markup—via

```
$$\vcenter{\noindent\sevenrm\csize=3ex
\puzzletrue\crw\data\parindent=3ex
\vtop{\hsize=21ex\obeylines Across
2 Switch mode
3 Knuth
6 Prior to \TeX}
\vtop{\hsize=25ex\obeylines Down
1 Public domain
2 All right
4 All comes to it
5 Atari type}
\fiverm\csize=3ex
\puzzlefalse\def\num{}\crw\data}$$
```

The automatic placement of crosswords in the text, as ‘floating bodies,’ is not dealt with. See therefore the general `\figplace.tex`, Zalmstra and Rogers (1989).

⁷This revealed a driver bug. `\hbox to \csize {...\cc}` in the crossword was not completely black, compare .

⁸Jürgen Knappen communicated the elegant solution `\ifnum`#1 = \lccode`#1...` Victor Eijkhout suggested among others `\ifnum \sfcode`#1 = 1000...` (T_EXbook p.76). The elegant solution is also in his book.

5 Variations

5.1 Black or . . .

The appearance of the black cells can be changed via redefinition of `\cc`.

Solutions are also typeset just with letters at the appropriate places in the diagram; without rules and black cells.

```

P   O N
D E K
   N   S
E D I T

```

This can be obtained after `\input crw.tex`, data definition and markup, via

```

\def\sol#1{\vtop{\offinterlineskip%
\halign{\&\hbox to\csize{\vrule%
height.8\csize depth.2\csize%
width0pt\proc##}\cr%end preamble
#1}}}%end \sol
\def\proc#1{\if*#1\else\ifx\relax#1%
\else%no black cells
\hfil\uppercase{#1}\fi\fi\hfil}%end\proc
$$\def\rs{\cr}\sol\data$$

```

5.2 No solution diagrams

Why not supply the solution similar to the clues, as columns of text?

Across	Down
2 ON	1 PD
3 DEK	2 OK
6 EDIT	4 END
	5 ST

5.3 *-symbol

It is a minor adaptation to use another symbol to denote crossed out cells.

6 Conclusion

Is this just for FUN? I hope that the approach facilitates the use of T_EX for typesetting crosswords. Moreover, the worked out ideas for inputting data WYSIWYG-like and have T_EX to modify them appropriately for further processing, can be applied to typesetting of other games, for example bridge and chess, and in general to typesetting of (simple) tables with natural input.

It was easier to write the macro from scratch than to understand the L^AT_EX macros of Hamilton Kelley. To my experience this is symptomatic for L^AT_EX, and its derivatives.

From the application point of view the following general issues have been dealt with

- automatic mark up
- natural (and minimal) input
- automatic and hidden numbering

- ruled tables.

T_EXnically use has been made of

- filling it up (`\leaders`, `\hrule`)
- testing case of letter (`\ifnum`#1 = \lccode`#1`)
- *local* catcode changes for making carriage return active (`\catcode`^^M= 13`)⁹
- 2-part macros and storing the data provided in between by the user
- abstraction from `&` and `\cr`
- parameterizing via a switch and redefs
- testing for empty argument (`\ifx \empty #2 \empty . . .`)
- nested if-s
- replicator mechanism in preamble
- parameter separation
- FIFO recursion.

The author claims that crosswords can be typeset easily and of high-quality via T_EX.

All the nitty-gritties put in reminded me of D_EK¹⁰

‘Yet dozen small refinements add up to something that is important to me, and I think such refinements might prove important to other people as well.’

7 Acknowledgements

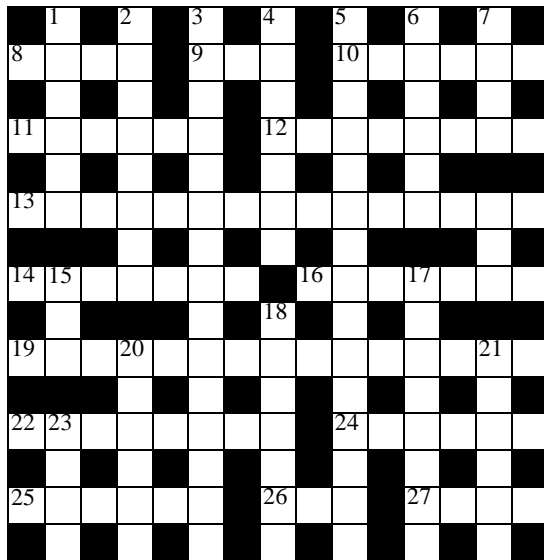
I like to thank Jürgen Knappen for providing the elegant test for the case of a letter. David Salomon is kindly acknowledged for the stimulating (e-mail) discussions.

References

- [1] Hamilton Kelley, B (1990): Some macros to draw crosswords. *TUGboat* 11, no. (1), 103–119.
- [2] Knuth, D.E (1986): *The T_EXbook*. Addison-Wesley.
- [3] Laan, C.G. van der (submitted for publication): FIFO and LIFO incognito.
- [4] Salomon, D (priv.comm.).
- [5] Zalmstra, J, D.F. Rogers (1989): A page make-up macro. *TUGboat* 10, no. (1), 73–81.

⁹Also touched upon is to allow spaces for *’s, except at the line ends. This entailed making spaces active.

¹⁰Mathematical Typography, Bulletin of the AMS, 1, 2, 1979, p.345.

AppendixHamilton Kelley's puzzle¹¹

is obtained—after `\input crw.tex`—via

```
\bdata%BHK's example
S I C T D S P*
Swam Oho Icecap
o p m r t n l*
```

```
Bopeep Schedule
s l a i y u *
Thalassographer
e s n a r*
HAirpin UmbRage
r o S b i *
ScaLenetriangLe
o a u c g e*
AMounted Allege
a v e e l e a*
Floral Nil Tace
l e y t y s y*
\edata\markup\data
$$\puzzletrue\crw\data$$
```

The above shows that the macros also take spaces for crossed out cells, except for those in the last column. It seems to be impossible to get hold of spaces at line ends in \TeX , because \TeX deletes endspaces, to accommodate for some computer series which fill up lines with trailing spaces. I could circumvent the S-state, where multiple spaces are skipped, \TeX book p.46.

The above is the next best to complete natural input for crosswords. If the spaces interfere with other parts of the document, the last line `{\obeyspaces...}`, of `crw.tex` can be deleted, and no spaces are accounted for in the crossword.

¹¹ Without BHK's clues, see for those his article.