

The sudoku bundle for displaying, solving and generating Sudoku puzzles*

Peter Wilson[†]
The Herries Press

2006/02/19

Abstract

The `sudoku bundle` provides a coordinated set of packages for displaying, solving, and generating Sudoku puzzles. A set of over 50 puzzles is also supplied.

Contents

1	Introduction	2
2	Usage	2
2.1	Typesetting	4
2.2	Solving	5
2.3	Generation	6
3	Code	8
3.1	The <code>printsudoku</code> package	8
3.2	The <code>solvesudoku</code> package	11
3.2.1	Setup	11
3.2.2	Binary solution sets	13
3.2.3	Problem initialisation	19
3.3	Solution methods	21
3.3.1	Checking the solution	40
3.4	The <code>createsudoku</code> package	45
4	Example sudoku puzzles	52

*This file (`sudokubundle.dtx`) has version number 1.0a, last revised on 2006/02/19.

[†]`herries dot press at earthlink dot net`

List of Figures

1	Example Sudoku puzzle	3
2	A sudoku file for the example puzzle	3
3	Solution to example puzzle	4

1 Introduction

In December 2005 the *PracTeX Journal* [PJ05] set a competition about Sudoku puzzles. Depending on their experience with \TeX contestants were asked to (a) typeset a particular puzzle, (b) typeset a puzzle described in a ‘sudoku’ file, (c) create a solver for sudoku puzzles. I entered the competition with a printer and solver. Following from this it was no great effort to develop a matching Soduko puzzle generator.

The packages described here have been developed completely independently from Paul Abraham’s `sudoku` package [Abr05], which I did not see until after I had finished work on them.

A Sudoku puzzle consists a 9 by 9 grid of cells with some of the cells containing a number between 1 and 9, such as is shown in Figure 1. The problem is to place a number between 1 and 9 in each cell such that no number appears more than once in each row and in each column and in each minor 3 by 3 grid. The solution to the example puzzle is shown in Figure 3. The puzzle and answer have been typeset using the `printsudoku` package.

Among many other sources the *Sudoku Solver by logic* website [SSBL] provides much information on Sudoku puzzles and their solutions, as does the *Sudoku Online* [SOL] website.

A Sudoku puzzle may be represented as a simple text file consisting of nine rows of numbers and dots, nine numbers and dots in each row. The numbers are the clues to the puzzle and the dots represent blanks in the grid. A sudoku file for the example puzzle is given in Figure 2.

This manual is typeset according to the conventions of the \LaTeX `DOCSTRIP` utility which enables the automatic extraction of the \LaTeX macro source files [MG04].

2 Usage

The bundle provides three packages; one for typesetting a puzzle, one for solving a puzzle, and the third for generating a puzzle. Each package requires the previous one(s).

A file is also provided that contains over 50 puzzles and their answers — all you have to do is run `LaTeX` on it, print it, and get to work on the puzzles (or give them to someone else).

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

Figure 1: Example Sudoku puzzle

```

..483..72
.12...8.
..52.13..
....62.91
7..5.9..3
94.78....
..39.74..
.5...61.
.8..469..

```

Example puzzle
(anything can come after the nine puzzle lines)

Figure 2: A sudoku file for the example puzzle

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

Figure 3: Solution to example puzzle

2.1 Typesetting

The `printsudoku` package enables you to typeset a Sudoku puzzle given in a `sudoku` file.

`\sudoku` The command `\sudoku{<file>}` typesets the sudoku puzzle from the `sudoku` file named `<file>`. For example:

```
\begin{center} \sudoku{puzzle1.sud} \end{center}
```

The command reads only the first nine lines in the file, which must contain the puzzle's description. Anything after these lines is ignored, so comments, such as a title or other explanatory text, can be put at the end of the file.

`\cluefont` Following `\cluefont{}` the clues are typeset using the `` font specification. The default is:

```
\cluefont{\Huge}
```

`\cellsize` The puzzle is typeset as a \LaTeX picture and `\cellsize{<length>}` specifies the size of the cells in the grid. To match the `\cluefont` the default is:

```
\cellsize{2.5\baselineskip}
```

Figure 1 was set using the default `\cluefont` and `\cellsize`. On the other hand, Figure 3 was set via:

```
\begin{figure}
\centering
\cluefont{\normalsize}\cellsize{1.5\baselineskip}
\sudoku{examout.ans}
\caption{Solution to example puzzle}\label{fig:ans1}
\end{figure}
```

`\writepuzzle` The command `\writepuzzle`, which takes nine required arguments each of which is a line in the puzzle, writes the puzzle to the file `\puzzlefile`, where the default is:

```
\newcommand*{\puzzlefile}{puzz.sud}
```

The nine required arguments to `\writepuzzle` are the lines describing a puzzle as they would appear in a sudoku file. There is an optional tenth argument after the nine required arguments which you may use to write a comment at the end of the file. For instance, earlier I used the following to write out the example puzzle to the file `examout.sud`:

```
\renewcommand*{\puzzlefile}{examout.sud}
\writepuzzle%
{.483..72}%
{.12...8.}%
{.52.13..}%
{...62.91}%
{7..5.9..3}%
{94.78...}%
{.39.74..}%
{.5...61.}%
{8..469..}%
[Example puzzle \sudpuzznewline
 (anything can come after the nine puzzle lines)]
```

And similarly for writing out the file with the solution.

If you want a multiline comment, use the `\sudpuzznewline` macro to start a new line (`\` or `\newline` will not work).

2.2 Solving

The `solvesudoku` package lets you use L^AT_EX to try and solve a Sudoku puzzle.

`\sudokusolve` The macro `\sudokusolve{<file>}` attempts to find a solution to the puzzle in the sudoku file `<file>`. It uses the `printsudoku` package for typesetting the puzzle.

It first prints the puzzle as given and then attempts to solve it. The solution, or as much as was obtainable, is typeset in a smaller font below the original. Here is the definition:

```
\newcommand*{\sudokusolve}[1]{%
  \begin{center}
    \sudoku{#1}
  \end{center}
  \getproblem{#1}
  \reduceallcells \keepreducing
  \writegame
  \noindent\begin{minipage}{\linewidth}
  THE ANSWER
  \begin{center}
    \cluefont{\normalsize}
    \cellsize{1.5\baselineskip}
    \sudoku{sud.out}
  \end{center}
  \end{minipage}}
```

You may use the macros in `\sudokusolve` separately if you wish.

<code>\getproblem</code>	<code>\getproblem{<file>}</code> reads in the puzzle from the sudoku file <code><file></code> and initialises the solution.
<code>\reduceallcells</code>	<code>\reduceallcells</code> takes a simple-minded approach to develop a solution to the puzzle.
<code>\keepreducing</code>	If <code>\reduceallcells</code> does not completely solve the puzzle then <code>\keepreducing</code> applies increasingly sophisticated solution methods, which may or may not lead to a complete solution.
<code>\writegame</code> <code>\sudsolnfile</code>	<code>\writegame</code> writes out a (partially) solved puzzle to the sudoku file <code>\sudsolnfile</code> (default <code>sud.out</code>). Later this is read in again by <code>\sudoku</code> to typeset the solution.

Here's a small example of how you might use the package. This will keep asking on the terminal for sudoku files to solve.

```
%%% solvethem.tex   Solve Sudoku files
%%%               Peter Wilson
\documentclass{article}
\usepackage{solvesudoku}
\newcommand*{\solvefile}[1]{\begingroup
  \sudokusolve{#1}%
  \par
  \vspace{\baselineskip}
  Number of clues = \the\numcluesctr\ and difficulty = \the\difficultyctr.
\endgroup}
\def\yesans{y}
\begin{document}
\loop
  \typein[\getans]{New file? y/n}
  \ifx\yesans\getans
    \typein[\sudfile]{Enter the file name}
    \clearpage
    \begin{center} \Huge \sudfile \end{center}
    \solvefile{\sudfile}
  \repeat
\end{document}
```

<code>\numcluesctr</code> <code>\difficultyctr</code>	These are two TeX counts. <code>\numcluesctr</code> is the number of clues in the initial puzzle, and <code>\difficultyctr</code> is a numerical indication of how hard the solver had to work, but it doesn't really mean much as the solver has had successes with some puzzles with a high <code>\difficultyctr</code> rating but failed to solve some with medium values of <code>\difficultyctr</code> .
--	---

2.3 Generation

With the `createsudoku` package you can get L^AT_EX to automatically generate a Sudoku puzzle. This package does require the use of Donald Arseneau's random code [Ars95] for generating random numbers.

<code>\generategrid</code>	<code>\generategrid[<file>]</code> will generate a Sudoku puzzle starting with the solution
----------------------------	---

given in the sudoku file *(file)*. If the file is not given then it will start from a solution provided by the package.

The puzzle is generated by taking the solution and randomly exchanging rows and columns for some (random) number of times. Then clues are eliminated until it is no longer soluble by `\sudokusolve`. The last clue that was deleted is put back and then the resulting grid is presented as the puzzle.

An initial set of clues are deleted before the solver is called. It can happen that the puzzle even then is too difficult for the solver, so you have to check the solution as presented is a complete one for the puzzle.

`\genfile` The generated puzzle, and its solution, is typeset and the puzzle is also output
`\prevfile` in the sudoku file `\genfile`. The default is:
`\currfile` `\newcommand*{\genfile}{gensud.sud}`

Two temporary sudoku files, `\prevfile` and `\currfile`, are used during the generation. The defaults are:

`\newcommand*{\prevfile}{genprev.sdx}`
`\newcommand*{\currfile}{gencurr.sdx}`

A pre-existing version of any of these three files will be overwritten.

`\setsudrandom` The random number generator is initialised via a seed based on a number (the internal value of `\randomi`) in the range $-1 \leq \text{\randomi} \leq 2147483646$. If its value is -1 (set via `\setsudrandom{-1}`) then the actual seed will be calculated based on the time and date. For any given seed the sequence of the generated random numbers is the same. The value of the seed is output via a `\typeout` if you need to repeat a generation.

You should not attempt to set `\randomi` directly but use `\setsudrandom{<num>}`.

The default setting is:

`\setsudrandom{-1}`

and as that means that the seed depends on the time and date, a new puzzle will be generated each time (unless the time interval is very short). The maximum number of puzzles that can be generated from one initial solution is 2,147,483,646 which is the range of the random number generator.

`\initialelimination` The macro `\initialelimination` eliminates an initial set of solutions from the randomized starting grid. Various predefined sets are provided and you can either provide your own definition for `\initialelimination` or `\let` it to one of the predefined sets, e.g.,
`\let\initialelimination\elimcrossandnines`

`\elimseventeen` `\elimseventeen` randomly eliminates 17 solutions from the grid — apparently
`\elimnum` the solution to a puzzle can be ambiguous if two numbers are completely absent.
`\elimcross` `\elimnum{<num>}` eliminates every solution *(num)* (e.g. `\elimnum{7}` will elim-
`\elimex` inate every 7 from the grid). `\elimcross` eliminates all numbers from column 5
`\elimcrossandnines` and from row 5, while `\elimex` eliminates all numbers on the diagonals.

`\elimcrossandex` `\elimcrossandnines` is a combination of `\elimcross` and `\elimnum{9}`. Sim-
`\elimcrossandexandnines` ilarly the macros `\elimcrossandex` and `\elimcrossandexandnines` are combina-
tions of some of the earlier macros.

3 Code

3.1 The printsudoku package

```

1 (*print)
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{printsudoku}[2006/02/16 v1.0 typeset sudoku puzzles]
4

\s@dread We are going to be reading and writing some files.
\s@dwrite 5 \newread\s@dread
6 \newwrite\s@dwrite
7

\reads@dgame \reads@dgame{<file>} opens a sudoku file <file> for reading.
8 %%% open game file
9 \newcommand*\reads@dgame}[1]{%
10 \closein\s@dread
11 \openin\s@dread=#1}
12

\halfs@dcell We're going to use a picture environment for drawing the grid, and to
\cellsize keep numbers neatly we'll use the length \halfs@dcell as the \unitlength.
\cluefont \cellsize{<length>} is the use command for setting the size of a cell. The clues
are typeset using the \cluefont font.
13 %%% set size of a game cell
14 \newlength\halfs@dcell
15 \newcommand*\cellsize}[1]{%
16 \halfs@dcell=#1\relax
17 \halfs@dcell=0.5\halfs@dcell}
18 \cellsize{2.5\baselineskip}
19 %% set font for the numbers
20 \newcommand*\cluefont}[1]{\def\s@dfont{#1}}
21 \cluefont{\Huge}
22

\s@dncol Counts for looping and cell positions
\s@dnrow 23 %% counts for looping and cell positions
\s@dcolpos 24 \newcount\s@dncol
\s@drowpos 25 \newcount\s@dnrow
26 \newcount\s@dcolpos
27 \newcount\s@drowpos
28

\adds@dclues \adds@dclues inserts the clues into the game grid. It reads the sudoku file line
by line insert the clues into the grid for one line before going on to the next line.
29 %% insert the clues into the game array
30 \newcommand*\adds@dclues}{%
31 \firsts@dcluetrue

```

```

32 \s@dnrow\@ne
33 \loop
34 \ifnum\s@dnrow<10\relax
35 \read\s@dread to \s@dline
36 \firsts@dcluetrue
37 \s@drowpos=\s@dnrow
38 \multiply\s@drowpos\tw@
39 \advance\s@drowpos\m@ne
40 \s@dncol\@ne
41 \dos@dcols
42 \advance\s@dnrow\@ne
43 \repeat
44 \closein\s@dread}
45

```

`\dos@dcols` `\dos@dcols` inserts one row of clues into the game grid.

```

46 %% insert a row of clues
47 \newcommand*{\dos@dcols}{%
48 \bgroup
49 \loop
50 \ifnum\s@dncol<10\relax
51 \s@dcolpos=\s@dncol
52 \multiply\s@dcolpos\tw@
53 \advance\s@dcolpos\m@ne
54 \put(\s@dcolpos,-\s@drowpos){\makebox(0,0){\s@dfont\gets@dclue}}%
55 \advance\s@dncol\@ne
56 \repeat
57 \egroup}
58

```

`\gettwo` `\splitoff{string}` gets the next character in a string. The next character is made available as `\istchar` and the remainder of the string as `\restchars`. This is explained in a forthcoming *Glisterings* column in *TUGboat*.

```

\istchar
\restchars
\splitoff
59 %% get the next character in a string
60 \def\gettwo#1#2\nowt{%
61 \gdef\istchar{#1}\gdef\restchars{#2}}
62 \def\splitoff#1{\gettwo#1\nowt}

```

`\s@dfstop` We use this for checking if a character extracted from a string is a ‘.’.

```

63 %% a full stop (period)
64 \gdef\s@dfstop{.}
65

```

`\iffirsts@dclue` `\gets@dclue` gets the next clue (character) from a line of clues. We do slightly different things if we are dealing with the first clue or the others.

```

\iffirsts@dcluetrue
\firsts@dcluefalse
\firsts@dcluefalse
\gets@dclue
66 %% get the next clue in the line
67 \newif\iffirsts@dclue
68 \firsts@dcluetrue
69 \newcommand*{\gets@dclue}{%

```

```

70 \iffirsts@dclue
71   \expandafter\splitoff\expandafter{\s@dline}%
72   \global\firsts@dcluefalse
73 \else
74   \expandafter\splitoff\expandafter{\restchars}%
75 \fi
76 \ifx\s@dfstop\istchar
77 \else
78   \istchar
79 \fi}
80

```

`\s@dgame` `\s@dgame` typesets the grid, then adds in the clues.

```

81 %% typeset the grid, then add the clues
82 \newcommand*\s@dgame{%
83   \setlength\unitlength\half@dcell
84   \begin{picture}(18,18)(0,-18)
85     \thinlines
86     \multiput(0,0)(2,0){10}{\line(0,-1){18}}
87     \multiput(0,0)(0,-2){10}{\line(1,0){18}}
88     \thicklines \linethickness{1pt}
89     \multiput(0,0)(6,0){4}{\line(0,-1){18}}
90     \multiput(0,0)(0,-6){4}{\line(1,0){18}}
91     \adds@dclues
92   \end{picture}}
93

```

`\sudoku` `\sudoku{⟨file⟩}` reads a game from `⟨file⟩` and then typesets it.

```

94 %% the whole shebang, where #1 is the name of the game file
95 \newcommand*\sudoku[1]{%
96   \reads@dgame{#1}%
97   \s@dgame}
98

```

`\puzzlefile` `\writepuzzle` takes nine arguments — the nine lines specifying a puzzle — and

`\writepuzzle` writes them to the `\puzzlefile` file.

`\writes@dpuzzend` `\writes@dpuzzend[⟨comment⟩]` writes `⟨comment⟩` at the end of the `\puzzlefile` file.

`\sudpuzznewline`

`\sudpuzznewline` provides a `\newline` macro for use in a comment being written to a puzzle file.

```

99 \newcommand*\puzzlefile{puzz.sud}
100 \newcommand*\writepuzzle[9]{%
101   \immediate\closeout\s@dwrite
102   \immediate\openout\s@dwrite=\puzzlefile
103   \immediate\write\s@dwrite{#1}%
104   \immediate\write\s@dwrite{#2}%
105   \immediate\write\s@dwrite{#3}%
106   \immediate\write\s@dwrite{#4}%
107   \immediate\write\s@dwrite{#5}%

```

```

108 \immediate\write\s@dwrite{#6}%
109 \immediate\write\s@dwrite{#7}%
110 \immediate\write\s@dwrite{#8}%
111 \immediate\write\s@dwrite{#9}%
112 \writes@dpuzzend}
113 \newcommand*{\writes@dpuzzend}[1][\@empty]{
114 \ifx\@empty #1\else
115 \immediate\write\s@dwrite{ }%
116 \immediate\write\s@dwrite{#1}%
117 \fi
118 \immediate\closeout\s@dwrite}
119 \newcommand*{\sudpuzznewline}{^^J}
120

```

The end of the package.

```
121 </print>
```

3.2 The solvesudoku package

The package requires the printsudoku package to do some puzzle typesetting.

```

122 < *solve>
123 \NeedsTeXFormat{LaTeX2e}
124 \ProvidesPackage{solvesudoku}[2006/02/16 v1.0 solve sudoku puzzles]
125 \RequirePackage{printsudoku}
126

```

3.2.1 Setup

`\commentary` It can be useful to control the amount of output to the log file.

```

127 \newcommand{\commentary}[1]{\typeout{#1}}
128

```

The following counts are defined in the printsudoku package:

`\s@dncol`, `\s@dnrow`, `\s@dcolpos`, `\s@drowpos`

```

\s@lcnta A bunch of \counts. They could probably be reduced in number but they help
\s@lcnt me to keep the code less mysterious.
\tenscnt 129 %%% too many counts, maybe they should be reduced
\tempcntz 130 \newcount\s@lcnta % a cell number
\tempcnty 131 \newcount\s@lcnt % number of solved cells
\tmpsetctr 132 \newcount\settonumcnt
\tmpsetansctr 133 \newcount\tenscnt
\boxctr 134 \newcount\tempcntz
\sumctr 135 \newcount\tempcnty
\toprangectr 136 \newcount\tmpsetctr
137 \newcount\tmpsetansctr
138 \newcount\boxctr
139 \newcount\sumctr
140 \newcount\toprangectr

```

```

\lonecellctr
  \reducedctr 141 \newcount\lonecellctr
  \numlistctr 142 \newcount\reducedctr
\difficultyctr 143 \newcount\numlistctr
  \s@dttempcnta 144 \newcount\difficultyctr
  \s@dttempcntb 145 \newcount\s@dttempcnta
\s@dttemploopcnta 146 \newcount\s@dttempcntb
  \maxrangectr 147 \newcount\s@dttemploopcnta
  \asetctr 148 \newcount\maxrangectr
  \secondctr 149 \newcount\asetctr
  150 \newcount\secondctr

  \digitictr
  \digitiictr 151 \newcount\digitictr
  \numdigitsctr 152 \newcount\digitiictr
  \numcluesctr 153 \newcount\numdigitsctr
  154 \newcount\numcluesctr
  155

\iffirsts@dclue
  \iffirstchar 156 %% \iffirsts@dclue% defined in printsudoku package
\ifnotgotthechar 157 \newif\iffirstchar
  \ifsetchanged 158 \newif\ifnotgotthechar
  \ifchanged 159 \newif\ifsetchanged
  \ifanychanged 160 \newif\ifchanged
  \iflonerchanged 161 \newif\ifanychange
  \ifstilldigits 162 \newif\iflonerchanged
  \ifkeepon 163 \newif\ifstilldigits
  164 \newif\ifkeepon
  \ifpairchanged 165 \newif\ifpairchanged
  166

  \newknt \newknt{<name>} creates a new count \name, which can include alphabetic
  \useknt characters. \useknt{<name>} calls the \name count. These are like the LATEX
\createsudsets kernel \@namedef and \@nameuse. In particular we will use 81 ‘numbered’ counts,
  one for each cell in the sudoku grid, using \createsudsets to generate them.

  167 %% create and use a numbered count
  168 %% we’ll use 81 of these, one for each cell in the grid
  169 \newcommand*\newknt}[1]{\expandafter\newcount\csname #1\endcsname}
  170 \newcommand*\useknt}[1]{\csname #1\endcsname}
  171 %% create potential answer sets
  172 \newcommand*\createsudsets}{%
  173 \global\s@lcnta\@ne
  174 \global\s@lcnt\z@
  175 \loop
  176 \ifnum\s@lcnta<82\relax
  177 \newknt{s@lans\the\s@lcnta}%
  178 \advance\s@lcnta\@ne
  179 \repeat}

```

```
180 \createsudsets
181
```

3.2.2 Binary solution sets

The major concern was deciding on the datastructure for the problem (the value of `\useknt{\s@lans\the\s@lcnta}` for the `\s@lcnta` cell). In the end I used a 9 digit ‘binary solution set’ for the potential solutions for each cell. (`[111111111]` \leftrightarrow 123456789 and `[101010101]` \leftrightarrow 13579). A cell solution, say `N`, is represented as the ‘set’ `-N` (e.g., a potential solution 3 is represented as `[001000000]` and the actual solution 3 is represented by `[-3]`).

```
\initialisesuddata This sets each of the 81 cells to 11111111.
182 %% initialise potential answer sets
183 \newcommand*\initialisesuddata{\bgroup
184 \global\s@lcnta@ne
185 \global\solcnt@z@
186 \global\difficultyctr@z@
187 \global\numcluesctr@z@
188 \loop
189 \ifnum\s@lcnta<82\relax
190 \global\useknt{s@lans\the\s@lcnta}=11111111\relax
191 \advance\s@lcnta@ne
192 \repeat
193 \egroup}
194

195 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% utility macros for ‘binary solution sets’ %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
196

\settonum \settonum{<set>}{<cnt>} converts a potential solution binary set <set> to the cor-
responding set of digits; that is [11...1]  $\rightarrow$  12...9. The result of the con-
version is assigned to the \count <cnt> which must be supplied by the calling
macro. If the set is negative then the result is that number (e.g. [-3]  $\rightarrow$  -3). If
the set contains only a single non-zero entry, that is converted to the negative of
the corresponding digit (e.g. [100]  $\rightarrow$  -7).

197 %\settonum{110011...}{cnt}
198 %% converts a potential solution set to a number ([11...1]  $\rightarrow$  12...9)
199 %% returns the input if it is negative ([-3]  $\rightarrow$  -3)
200 %% returns a negative number if the set represents a single digit ([10]  $\rightarrow$  -8)
201 \newcommand*\settonum[2]{%
202 \settonumcnt=#1\relax
203 \tempcnty=\z@
204 \tenscnt=\@ne
205 \ifnum\settonumcnt<\z@ % just return the number
206 \tempcnty=\settonumcnt
207 #2=\tempcnty
208 \else
209 \ifodd\settonumcnt % 9 flagged
```

```
210     \tempcntz=9\relax
211     \multiply\tempcntz \tenscnt
212     \advance \tempcnty by \tempcntz
213     \multiply\tenscnt 10\relax
214 \fi
215 \divide\settonumcnt by 10\relax
216 \ifodd\settonumcnt % 8 flagged
217     \tempcntz=8\relax
218     \multiply\tempcntz \tenscnt
219     \advance \tempcnty \tempcntz
220     \multiply\tenscnt 10\relax
221 \fi
222 \divide\settonumcnt by 10\relax
223 \ifodd\settonumcnt % 7 flagged
224     \tempcntz=7\relax
225     \multiply\tempcntz \tenscnt
226     \advance \tempcnty \tempcntz
227     \multiply\tenscnt 10\relax
228 \fi
229 \divide\settonumcnt by 10\relax
230 \ifodd\settonumcnt % 6 flagged
231     \tempcntz=6\relax
232     \multiply\tempcntz \tenscnt
233     \advance \tempcnty \tempcntz
234     \multiply\tenscnt 10\relax
235 \fi
236 \divide\settonumcnt by 10\relax
237 \ifodd\settonumcnt % 5 flagged
238     \tempcntz=5\relax
239     \multiply\tempcntz \tenscnt
240     \advance \tempcnty \tempcntz
241     \multiply\tenscnt 10\relax
242 \fi
243 \divide\settonumcnt by 10\relax
244 \ifodd\settonumcnt % 4 flagged
245     \tempcntz=4\relax
246     \multiply\tempcntz \tenscnt
247     \advance \tempcnty \tempcntz
248     \multiply\tenscnt 10\relax
249 \fi
250 \divide\settonumcnt by 10\relax
251 \ifodd\settonumcnt % 3 flagged
252     \tempcntz=3\relax
253     \multiply\tempcntz \tenscnt
254     \advance \tempcnty \tempcntz
255     \multiply\tenscnt 10\relax
256 \fi
257 \divide\settonumcnt by 10\relax
258 \ifodd\settonumcnt % 2 flagged
259     \tempcntz=2\relax
```

```

260     \multiply\tempcntz \tenscnt
261     \advance \tempcnty \tempcntz
262     \multiply\tenscnt 10\relax
263     \fi
264     \divide\settonumcnt by 10\relax
265     \ifodd\settonumcnt % 1 flagged
266     \tempcntz=1\relax
267     \multiply\tempcntz \tenscnt
268     \advance \tempcnty \tempcntz
269 %%     \multiply\tenscnt 10\relax
270     \fi
271     \ifnum \tempcnty<10
272     \ifnum\tempcnty>\z@
273     \tempcnty = -\tempcnty
274     \fi
275     \fi
276     #2=\tempcnty
277 \fi}
278

```

`\numofnuminset` `\numofnuminset{<digit>}{<set>}{<cnt>}` set the `\count` `<cnt>` to the number of times the digit `<digit>` is represented in the binary set `<set>`. For example, the number of digits in the set [200000013] are 1 -> 2, 2 -> 0, ..., 8 -> 1 and 9 -> 3.

```

279 %%\numofnuminset{digit}{set}{returnctr}
280 %% returns the number of times digit is represented in the set
281 %% e.g. if the set is [200000013] then 1 -> 2, 2 -> 0,... 8 -> 1, 9 -> 3
282 \newcommand*{\numofnuminset}[3]{%
283   \tmpsetctr=#2\relax
284   \tmpsetansctr=\tmpsetctr
285   \ifnum\tmpsetctr<\z@% solution, not a set
286     \tmpsetansctr=\z@
287   \else
288     \ifcase #1\relax
289     \or % 1
290       \divide\tmpsetansctr by 10000000\relax
291     \or % 2
292       \divide\tmpsetansctr by 1000000\relax
293       \tmpsetctr=\tmpsetansctr
294       \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
295       \advance\tmpsetansctr -\tmpsetctr
296     \or % 3
297       \divide\tmpsetansctr by 100000\relax
298       \tmpsetctr=\tmpsetansctr
299       \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
300       \advance\tmpsetansctr -\tmpsetctr
301     \or % 4
302       \divide\tmpsetansctr by 10000\relax
303       \tmpsetctr=\tmpsetansctr
304       \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax

```

```

305     \advance\tmpsetansctr -\tmpsetctr
306 \or           % 5
307     \divide\tmpsetansctr by 10000\relax
308     \tmpsetctr=\tmpsetansctr
309     \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
310     \advance\tmpsetansctr -\tmpsetctr
311 \or           % 6
312     \divide\tmpsetansctr by 1000\relax
313     \tmpsetctr=\tmpsetansctr
314     \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
315     \advance\tmpsetansctr -\tmpsetctr
316 \or           % 7
317     \divide\tmpsetansctr by 100\relax
318     \tmpsetctr=\tmpsetansctr
319     \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
320     \advance\tmpsetansctr -\tmpsetctr
321 \or           % 8
322     \divide\tmpsetansctr by 10\relax
323     \tmpsetctr=\tmpsetansctr
324     \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
325     \advance\tmpsetansctr -\tmpsetctr
326 \or           % 9
327     \tmpsetctr=\tmpsetansctr
328     \divide\tmpsetctr 10\relax \multiply\tmpsetctr 10\relax
329     \advance\tmpsetansctr -\tmpsetctr
330 \else         % error
331     \tmpsetansctr=\z@
332 \fi
333 \fi
334 #3=\tmpsetansctr}
335

```

`\deletenumfromset` `\deletenumfromset{⟨digit⟩}{⟨set⟩}{⟨cnt⟩}` removes the `⟨digit⟩` from the `⟨set⟩` and sets the `\count` `⟨cnt⟩` to the resulting modified set. If `⟨digit⟩` was the only member then the result is that digit negated. For example:

9 from [...11] -> [...10]

8 from [0...010] -> -8

```

336 %%\deletenumfromset{digit}{set}{returnctr}
337 %% deletes num from the [set], returning the modified set or a -ve number if
338 %% the digit was the only member.
339 %% e.g., 9 from [...11] -> [...10], 8 from [0...010] -> -8
340 \newcommand*\deletenumfromset}[3]{
341   \global\setchangedfalse
342   \tmpsetctr=#2\relax
343   \tmpsetansctr=#2\relax
344   \ifnum\tmpsetctr<\z@ % represents a solved number, do nothing
345 \else
346   \ifcase #1\relax
347     \or           % 1

```

```

348 \divide\tmpsetctr by 10000000\relax
349 \ifodd\tmpsetctr% it's there
350 \advance\tmpsetansctr -10000000\relax
351 \global\setchangedtrue
352 \fi
353 \or % 2
354 \divide\tmpsetctr by 1000000\relax
355 \ifodd\tmpsetctr% it's there
356 \advance\tmpsetansctr -1000000\relax
357 \global\setchangedtrue
358 \fi
359 \or % 3
360 \divide\tmpsetctr by 100000\relax
361 \ifodd\tmpsetctr% it's there
362 \advance\tmpsetansctr -100000\relax
363 \global\setchangedtrue
364 \fi
365 \or % 4
366 \divide\tmpsetctr by 10000\relax
367 \ifodd\tmpsetctr% it's there
368 \advance\tmpsetansctr -10000\relax
369 \global\setchangedtrue
370 \fi
371 \or % 5
372 \divide\tmpsetctr by 1000\relax
373 \ifodd\tmpsetctr% it's there
374 \advance\tmpsetansctr -1000\relax
375 \global\setchangedtrue
376 \fi
377 \or % 6
378 \divide\tmpsetctr by 100\relax
379 \ifodd\tmpsetctr% it's there
380 \advance\tmpsetansctr -100\relax
381 \global\setchangedtrue
382 \fi
383 \or % 7
384 \divide\tmpsetctr by 10\relax
385 \ifodd\tmpsetctr% it's there
386 \advance\tmpsetansctr -10\relax
387 \global\setchangedtrue
388 \fi
389 \or % 8
390 \divide\tmpsetctr by 1\relax
391 \ifodd\tmpsetctr% it's there
392 \advance\tmpsetansctr -1\relax
393 \global\setchangedtrue
394 \fi
395 \or % 9
396 \ifodd\tmpsetctr% it's there
397 \advance\tmpsetansctr -1\relax

```



```

444 \settonum{\the\useknt{s@lans38}}{\s@dttempcnta}\the\s@dttempcnta &
445 \settonum{\the\useknt{s@lans39}}{\s@dttempcnta}\the\s@dttempcnta &
446 \settonum{\the\useknt{s@lans40}}{\s@dttempcnta}\the\s@dttempcnta &
447 \settonum{\the\useknt{s@lans41}}{\s@dttempcnta}\the\s@dttempcnta &
448 \settonum{\the\useknt{s@lans42}}{\s@dttempcnta}\the\s@dttempcnta &
449 \settonum{\the\useknt{s@lans43}}{\s@dttempcnta}\the\s@dttempcnta &
450 \settonum{\the\useknt{s@lans44}}{\s@dttempcnta}\the\s@dttempcnta &
451 \settonum{\the\useknt{s@lans45}}{\s@dttempcnta}\the\s@dttempcnta \\ \hline
452 \settonum{\the\useknt{s@lans46}}{\s@dttempcnta}\the\s@dttempcnta &
453 \settonum{\the\useknt{s@lans47}}{\s@dttempcnta}\the\s@dttempcnta &
454 \settonum{\the\useknt{s@lans48}}{\s@dttempcnta}\the\s@dttempcnta &
455 \settonum{\the\useknt{s@lans49}}{\s@dttempcnta}\the\s@dttempcnta &
456 \settonum{\the\useknt{s@lans50}}{\s@dttempcnta}\the\s@dttempcnta &
457 \settonum{\the\useknt{s@lans51}}{\s@dttempcnta}\the\s@dttempcnta &
458 \settonum{\the\useknt{s@lans52}}{\s@dttempcnta}\the\s@dttempcnta &
459 \settonum{\the\useknt{s@lans53}}{\s@dttempcnta}\the\s@dttempcnta &
460 \settonum{\the\useknt{s@lans54}}{\s@dttempcnta}\the\s@dttempcnta \\ \hline\hline
461 \settonum{\the\useknt{s@lans55}}{\s@dttempcnta}\the\s@dttempcnta &
462 \settonum{\the\useknt{s@lans56}}{\s@dttempcnta}\the\s@dttempcnta &
463 \settonum{\the\useknt{s@lans57}}{\s@dttempcnta}\the\s@dttempcnta &
464 \settonum{\the\useknt{s@lans58}}{\s@dttempcnta}\the\s@dttempcnta &
465 \settonum{\the\useknt{s@lans59}}{\s@dttempcnta}\the\s@dttempcnta &
466 \settonum{\the\useknt{s@lans60}}{\s@dttempcnta}\the\s@dttempcnta &
467 \settonum{\the\useknt{s@lans61}}{\s@dttempcnta}\the\s@dttempcnta &
468 \settonum{\the\useknt{s@lans62}}{\s@dttempcnta}\the\s@dttempcnta &
469 \settonum{\the\useknt{s@lans63}}{\s@dttempcnta}\the\s@dttempcnta \\ \hline
470 \settonum{\the\useknt{s@lans64}}{\s@dttempcnta}\the\s@dttempcnta &
471 \settonum{\the\useknt{s@lans65}}{\s@dttempcnta}\the\s@dttempcnta &
472 \settonum{\the\useknt{s@lans66}}{\s@dttempcnta}\the\s@dttempcnta &
473 \settonum{\the\useknt{s@lans67}}{\s@dttempcnta}\the\s@dttempcnta &
474 \settonum{\the\useknt{s@lans68}}{\s@dttempcnta}\the\s@dttempcnta &
475 \settonum{\the\useknt{s@lans69}}{\s@dttempcnta}\the\s@dttempcnta &
476 \settonum{\the\useknt{s@lans70}}{\s@dttempcnta}\the\s@dttempcnta &
477 \settonum{\the\useknt{s@lans71}}{\s@dttempcnta}\the\s@dttempcnta &
478 \settonum{\the\useknt{s@lans72}}{\s@dttempcnta}\the\s@dttempcnta \\ \hline
479 \settonum{\the\useknt{s@lans73}}{\s@dttempcnta}\the\s@dttempcnta &
480 \settonum{\the\useknt{s@lans74}}{\s@dttempcnta}\the\s@dttempcnta &
481 \settonum{\the\useknt{s@lans75}}{\s@dttempcnta}\the\s@dttempcnta &
482 \settonum{\the\useknt{s@lans76}}{\s@dttempcnta}\the\s@dttempcnta &
483 \settonum{\the\useknt{s@lans77}}{\s@dttempcnta}\the\s@dttempcnta &
484 \settonum{\the\useknt{s@lans78}}{\s@dttempcnta}\the\s@dttempcnta &
485 \settonum{\the\useknt{s@lans79}}{\s@dttempcnta}\the\s@dttempcnta &
486 \settonum{\the\useknt{s@lans80}}{\s@dttempcnta}\the\s@dttempcnta &
487 \settonum{\the\useknt{s@lans81}}{\s@dttempcnta}\the\s@dttempcnta \\ \hline\hline
488 \end{tabular}}
489

```

3.2.3 Problem initialisation

```

490 %%% get the problem and initialise the solution %%%

```

491

`\getproblem` `\getproblem{⟨file⟩}` initialises the solution grid and reads the puzzle from the sudoku file `⟨file⟩`. It calls `\initialsoln` to read the file and insert the clues into the grid.

```
492 %% read the problem and initialise the problem (solution)
493 \newcommand*{\getproblem}[1]{\bgroup
494   \initialisesuddata
495   \global\solcnt\z@
496   \immediate\closein\s@dread
497   \immediate\openin\s@dread=#1
498   \initialsoln
499   \typeout{Sudoku problem #1, \the\solcnt\space initial clues}
500 \egroup}
501
```

`\initialsoln` Get the clues from the puzzle file, line by line, and insert them into the grid. `\s@dline` holds one line from the file.

```
502 %% insert the clues from the input game file
503 \newcommand*{\initialsoln}{%
504   \firsts@dcluetrue
505   \s@dnrow\@ne
506   \s@lcnta\@ne
507   \loop
508     \ifnum\s@dnrow<10\relax
509     \read\s@dread to \s@dline
510     \firsts@dcluetrue
511     \s@dncol\@ne
512     \dos@dinitialcols
513     \advance\s@dnrow\@ne
514   \repeat
515   \immediate\closein\s@dread}
516
```

`\dos@dinitialcols` Insert the clues from one line of the puzzle file into one row of the grid.

```
517 %% insert the clues from one line of the game file
518 \newcommand*{\dos@dinitialcols}{\bgroup
519   \loop
520     \ifnum\s@dncol<10\relax
521     \fixentry
522     \advance\s@dncol\@ne
523   \repeat
524 \egroup}
525
```

`\fixentry` Inserts a clue from the current line into the grid.

```
526 %% insert a clue from the current line of the game file
527 \newcommand*{\fixentry}{%
528   \iffirsts@dclue
```

```

529 \expandafter\splitoff\expandafter{\s@dline}%
530 \global\firsts@dccluefalse
531 \else
532 \expandafter\splitoff\expandafter{\restchars}%
533 \fi
534 \ifx\s@dfstop\istchar
535 \else
536 \global\useknt{s@lans\the\s@lcnta}=-\istchar
537 \global\advance\solcnt\@ne
538 \global\advance\numcluesctr\@ne
539 \fi
540 \global\advance\s@lcnta\@ne}
541
542

```

3.3 Solution methods

Three facts are used to generate a solution.

1. A digit must be unique within a row, a column, and a block. Thus, if a solution, say N , is known for a cell, then the potential solution N must be deleted from all cells in the row, column, and block that the solved cell is in. I have called this a simple reduction. The simple reductions are repeated until there are no changes to the potential solutions.
2. If among all the cells in a row (column, block) there is a digit that occurs only once among all the potential solutions, then that digit is the solution for its cell. I have termed this a loner. If a loner solution is found then simple reductions are repeated.
3. If among all the cells in a row (column, block) there are two digits which occur twice only in the row, each time as a pair (e.g., 39 and 39) then one or other of the two digits must be a solution for a cell in which the pairs occur. This means that the two digits in the pair cannot occur anywhere else in the row (column, block). If there are no loners then the problem is examined for pairs. If there is any change in a potential solution then the simple reduction process is applied.

The process stops either when all 81 cells have been solved or there is no change in any potential solution after going through all the above reductions.

There are more sophisticated solution techniques that could be applied but they are also increasingly difficult to code in \TeX .

`\reduceacell` `\reduceacell` looks at the solution set for the current cell (the `\s@lcnta` cell) and if the cell has been solved then it deletes the solution digit from the row, column, and box in which the cell appears. This may result in new solutions for other cells.

Given a cell number N ($0 < N < 82$) and, using integer arithmetic, putting $M = (N - 1)/9$ then

- Cell number C at top of column: $C = N - 9M$
- Cell number R at left of row: $R = 9M + 1$
- Cell number b at top left of 3 by 3 box: $B = 27(M/3) + 3((N - 1 - 9M)/3) + 1$

```

543 %% Eliminate a solution digit from its row, column and box.
544 %% This may result in some new solutions.
545 \newcommand*{\reduceacell}{\bgroup
546   \ifnum\useknt{s@lans\the\s@lcnta}<\z@ % solved cell
547     \s@dttempcnta=-\useknt{s@lans\the\s@lcnta}
548 %% get the col, row and box numbers
549   \tempcnty=\s@lcnta % M in the formulae
550   \advance\tempcnty\m@ne \divide\tempcnty 9\relax
551   \tempcntz=\tempcnty \multiply\tempcntz 9\relax
552   \s@dcolpos=\s@lcnta \advance\s@dcolpos -\tempcntz % col
553   \s@drowpos\tempcntz \advance\s@drowpos\@ne % row
554   \boxctr=\tempcnty % box
555   \divide\boxctr\thr@@ \multiply\boxctr 27\relax
556   \tempcnty=\s@lcnta % reuse \tempcnty
557   \advance\tempcnty\m@ne \advance\tempcnty -\tempcntz
558   \divide\tempcnty\thr@@ \multiply\tempcnty \thr@@
559   \advance\boxctr\tempcnty \advance\boxctr \@ne
560 %% reduce the row, column, and the box
561   \reducerow{\the\s@drowpos}{\the\s@dttempcnta}
562   \reducecol{\the\s@dcolpos}{\the\s@dttempcnta}
563   \reducebox{\the\boxctr}{\the\s@dttempcnta}
564   \fi
565 \egroup}
566

```

`\reducerow` `\reducerow{<row>}{<digit>}` eliminates `<digit>` from the binary solution sets of the cells in the row starting at cell number `<row>`.

```

567 %% \reducerow{rowcellnumber}{digit}
568 %% simple reduction for digit in a row
569 \newcommand*{\reducerow}[2]{\bgroup
570   \s@dncol=#1\relax
571   \toprangectr=\s@dncol
572   \advance\toprangectr 9\relax
573   \bgroup
574   \loop
575     \ifnum\s@dncol<\toprangectr
576       \deletenumfromset{#2}{\the\useknt{s@lans\the\s@dncol}}{\reducedctr}
577       \ifsetchanged
578         \global\changedtrue
579         \settonum{\the\reducedctr}{\numlistctr}
580         \ifnum\numlistctr<10\relax
581           \global\advance\solcnt\@ne
582 \commentary{\the\solcnt\space solved
583             (\the\s@dncol\space = \the\numlistctr\space in row starting #1)}%

```

```

584     \global\useknt{s@lans\the\s@dncol}=\numlistctr
585     \else
586     \global\useknt{s@lans\the\s@dncol}=\reducedctr
587     \fi
588     \fi
589     \advance\s@dncol\@ne
590     \repeat
591     \egroup
592 \egroup}
593

```

`\reducecol` `\reducecol{<col>}{<digit>}` eliminates *<digit>* from the binary solution sets of the cells in the column starting at cell number *<col>*.

```

594 %% \reducecol{colcellnumber}{digit}
595 %% simple reduction of a digit in a column
596 \newcommand*\reducecol[2]{\bgroup
597   \s@dnrow=#1\relax
598   \toprangelctr=\s@dnrow
599   \advance\toprangelctr 73\relax
600   \bgroup
601   \loop
602     \ifnum\s@dnrow<\toprangelctr
603     \deletenumfromset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\reducedctr}
604     \ifsetchanged
605       \global\changedtrue
606       \settonum{\the\reducedctr}{\numlistctr}
607       \ifnum\numlistctr<10\relax
608         \global\advance\solcnt\@ne
609 \commentary{\the\solcnt\space solved
610   (\the\s@dnrow\space = \the\numlistctr\space in column starting #1)}%
611       \global\useknt{s@lans\the\s@dnrow}=\numlistctr
612     \else
613       \global\useknt{s@lans\the\s@dnrow}=\reducedctr
614     \fi
615     \fi
616     \advance\s@dnrow 9\relax
617   \repeat
618   \egroup
619 \egroup}
620

```

`\reducebox` `\reducebox{<box>}{<digit>}` eliminates *<digit>* from the binary solution sets of the cells in the box starting at cell number *<box>*.

```

621 %% \reducebox{boxcellnumber}{digit}
622 %% simple reduction of a digit in a box
623 \newcommand*\reducebox[2]{\bgroup
624   \s@dncol=#1\relax
625   \toprangelctr=\s@dncol
626   \advance\toprangelctr \thr@@

```

```

627 \bgroup
628 \loop
629   \ifnum\s@dncol<\toprangelctr
630   \s@dnrow=\s@dncol
631   \deletenumfromset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\reducedctr}
632   \ifsetchanged
633     \global\changedtrue
634     \settonum{\the\reducedctr}{\numlistctr}
635     \ifnum\numlistctr<10\relax
636       \global\advance\solcnt\@ne
637 \commentary{\the\solcnt\space solved
638   (\the\s@dnrow\space = \the\numlistctr\space in box starting #1)}%
639   \global\useknt{s@lans\the\s@dnrow}=\numlistctr
640   \else
641     \global\useknt{s@lans\the\s@dnrow}=\reducedctr
642   \fi
643 \fi
644 \advance\s@dnrow 9\relax
645 \deletenumfromset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\reducedctr}
646 \ifsetchanged
647   \global\changedtrue
648   \settonum{\the\reducedctr}{\numlistctr}
649   \ifnum\numlistctr<10\relax
650     \global\advance\solcnt\@ne
651 \commentary{\the\solcnt\space solved
652   (\the\s@dnrow\space = \the\numlistctr\space in box starting #1)}%
653   \global\useknt{s@lans\the\s@dnrow}=\numlistctr
654   \else
655     \global\useknt{s@lans\the\s@dnrow}=\reducedctr
656   \fi
657 \fi
658 \advance\s@dnrow 9\relax
659 \deletenumfromset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\reducedctr}
660 \ifsetchanged
661   \global\changedtrue
662   \settonum{\the\reducedctr}{\numlistctr}
663   \ifnum\numlistctr<10\relax
664     \global\advance\solcnt\@ne
665 \commentary{\the\solcnt\space solved
666   (\the\s@dnrow\space = \the\numlistctr\space in box starting #1)}%
667   \global\useknt{s@lans\the\s@dnrow}=\numlistctr
668   \else
669     \global\useknt{s@lans\the\s@dnrow}=\reducedctr
670   \fi
671 \fi
672 \advance\s@dncol \@ne
673 \repeat
674 \egroup
675 \egroup}
676

```

`\reduceallcells` `\reduceallcells` examines all 81 cells, eliminating solution digits from the binary solution sets of the other cells in the row, column, and box.

```

677 %% loop over every cell, removing solution digits from the potential sets
678 %% in the cell's row, column and box.
679 \newcommand*{\reduceallcells}{\bgroup
680   \ifnum\difficultyctr<\@ne
681     \global\difficultyctr\@ne
682   \fi
683   \s@lcnta\@ne
684   \global\changedfalse
685   \loop
686     \ifnum\s@lcnta<82\relax
687       \reduceacell
688       \advance\s@lcnta\@ne
689     \repeat
690 \egroup}
691

```

`\keepreducing` `\keepreducing` performs simple reductions on all the cells until either the puzzle is solved or no changes have been made. It then tries loner reductions, returning to simple reductions following any change. If no change has been made after loner reductions it then tries pair reductions. After any change it goes back to simple reductions. This iterative process continues until either the puzzle has been solved (`\solcnt = 81`) or no further changes can be made.

A measure of the difficulty of the puzzle could be made by incrementing a count each time a new kind of reduction is tried. The increment increases with the complexity of the reduction.

```

692 \newcommand*{\keepreducing}{%
693   \anychangetrue
694   \keepontrue
695   \loop
696     \ifkeepon
697       \keepreducingcells
698       \ifnum\solcnt<81\relax
699   %%       \advance\difficultyctr 4\relax
700         \reduceloners
701       \fi
702       \anychangefalse
703       \iflonerchanged
704         \anychangetrue
705       \fi
706       \ifanychange\else
707         \ifnum\solcnt<81\relax
708   %%       \advance\difficultyctr 8\relax
709         \pairchangedfalse
710         \reducepairs
711         \ifpairchanged
712           \anychangetrue
713         \fi

```

```

714     \fi
715     \fi
716     \checkkeepon
717   \repeat
718   \ifnum\solcnt<81\relax
719     \global\multiply \difficultyctr 10\relax
720   \fi
721 \typeout{***** Number of solutions = \the\solcnt}
722 \typeout{***** Difficulty rating = \the\difficultyctr}
723 }
724

```

`\checkkeepon` Determines whether or not to keep on trying to reach the solution.

```

725 \newcommand*\checkkeepon{%
726   \ifnum\solcnt>80\relax
727     \keeponfalse
728   \fi
729   \ifanychange\else
730     \keeponfalse
731   \fi}
732

```

`\keepreducingcells` Keep on trying the simple reduction.

```

733 %% Keep performing the simple reduction
734 \newcommand*\keepreducingcells{\bgroup
735 \commentary{Simple reductions}%
736   \ifnum\difficultyctr<\tw@
737     \global\difficultyctr\tw@
738   \fi
739   \changedtrue
740   \loop
741     \ifchanged
742       \reduceallcells
743     \checksimplereductions
744   \repeat
745 \egroup}
746

```

`\checksimplereductions` Determines whether or not to keep on trying simple reductions.

```

747 \newcommand*\checksimplereductions{%
748   \ifchanged\else
749     \typesimplestatus
750   \fi
751   \ifnum\solcnt>80\relax%   solution found!
752     \changedfalse
753   \fi}
754

```

```

755 %%%%%%%%%%% reduce loners %%%%%%%%%%%
756

```

`\reduceloners` Reduce via the loner digits, trying rows first, and if no change then trying columns, and if still no change then trying boxes.

```

757 %% reduce the loner digits, per row, column and box
758 \newcommand*\reduceloners}{%
759 \commentary{reduceloners}%
760 \global\lonerchangedfalse
761 \ifnum\difficultyctr<4\relax
762 \global\difficultyctr=4\relax
763 \fi
764 \ifnum\solcnt>80\else
765 \bgroup
766 \reducerowloners
767 \iflonerchanged\else
768 \reducecolloners
769 \iflonerchanged\else
770 \reduceboxloners
771 \fi
772 \fi
773 \typelonestatus
774 \egroup
775 \fi}
776

```

`\reducerowloners` Loop through the rows and reduce the first loner found.

```

777 %% Loop through the rows, reducing loners
778 \newcommand*\reducerowloners}{%
779 \sumctr\z@
780 \s@dnrow\@ne
781 \loop
782 \ifnum\s@dnrow<10\relax
783 \s@lcnta=\s@dnrow \advance\s@lcnta\m@ne
784 \multiply\s@lcnta 9\relax \advance\s@lcnta\@ne
785 \sumrowsets
786 \ifnum\sumctr>\z@
787 \getloner{\the\sumctr}{\s@dtempcntb}
788 \ifnum\s@dtempcntb=\z@
789 \else
790 \reducelonerrowcell{\the\s@lcnta}{\the\s@dtempcntb}
791 \iflonerchanged
792 \advance\s@dnrow 10\relax
793 \fi
794 \fi
795 \fi
796 \advance\s@dnrow\@ne
797 \repeat}
798

```

`\sumrowsets` Add together all the solution binary sets for the unsolved cells in the current row (`\s@lcnta`). The result is in `\sumctr`.

```

799 %% add the sets in the current row
800 \newcommand*{\sumrowsets}{%
801   \bgroup
802   \s@dncol=\s@lcnta
803   \toprangectr=\s@dncol
804   \advance\toprangectr 9\relax
805   \global\sumctr\z@
806   \loop
807     \ifnum\s@dncol<\toprangectr
808       \ifnum\useknt{s@lans\the\s@dncol}>\z@
809         \global\advance\sumctr\useknt{s@lans\the\s@dncol}
810       \fi
811       \advance\s@dncol\@ne
812     \repeat
813   \egroup}
814

```

`\getloner` `\getloner{setsum}{cnt}` examines (the sum of) a solution set and returns the first loner (i.e., the first occurrence of the digit 1), or zero if there is none. The result is converted to a solution digit and assigned to the `\count` `<cnt>`.

For example, given the set [000200110] it will return 7 (the set indicates two 4s, one 7, and one 8).

```

815 %% get the loner from a setsum
816 %% \getloner{setsum}{returnctr} returns the (first) loner in a potential set,
817 %% or zero
818 \newcommand*{\getloner}[2]{\bgroup
819   \ifnum #1>\z@
820     \s@dttemploopcnta\@ne
821     \loop
822       \ifnum\s@dttemploopcnta<10\relax
823         \numofnuminset{\the\s@dttemploopcnta}{#1}{\s@dttempcnta}
824         \ifnum\s@dttempcnta=\@ne%           found a lone digit
825           \s@dttempcnta=\s@dttemploopcnta
826           \s@dttemploopcnta=100\relax
827         \fi
828         \advance\s@dttemploopcnta\@ne
829       \repeat
830       \ifnum\s@dttemploopcnta>99 \else \s@dttempcnta=\z@ \fi
831       \global#2=\s@dttempcnta
832     \else
833       \s@dttempcnta=\z@
834       \global#2=\s@dttempcnta
835     \fi
836   \egroup}
837

```

`\reducelonerowcell` `\reducelonerowcell{rowcell}{digit}`. Given (the cell at) the start of a row and a lone digit, find the cell in the row with that digit in its solution set, and make the digit the solution of that cell.

```

838 %% \reducelonerowcell{startofrowcell}{digit}
839 %% given a row and a lone digit, find the cell containing the digit
840 %% and make the digit a solution of that cell.
841 \newcommand*{\reducelonerowcell}[2]{\bgroup
842   \s@dncol=#1
843   \toprangectr=\s@dncol
844   \advance\toprangectr 9\relax
845   \ifnum #2<\@ne
846     \advance\s@dncol 100\relax
847   \fi
848   \loop
849     \ifnum\s@dncol<\toprangectr
850       \numofnuminset{#2}{\the\useknt{s@lans\the\s@dncol}}{\lonecellctr}
851       \ifnum\lonecellctr=\@ne%   this cell has the loner
852         \lonecellctr=#2
853         \global\useknt{s@lans\the\s@dncol}=-\lonecellctr
854         \global\lonerchangedtrue
855         \global\advance\solcnt\@ne
856 \commentary{\the\solcnt\space solved (#2 loner in row starting #1)}%
857         \s@dncol=\toprangectr
858       \fi
859     \advance\s@dncol\@ne
860   \repeat
861 \egroup}
862

```

`\reducecolloners` Loop through the columns and reduce the first loner found.

```

863 %% Loop through the columns, reducing loners
864 \newcommand*{\reducecolloners}{\bgroup
865   \sumctr\z@
866   \s@dncol\@ne
867   \loop
868     \ifnum\s@dncol<10\relax
869       \s@lcnta=\s@dncol
870       \sumcolsets
871       \ifnum\sumctr>\z@
872         \getloner{\the\sumctr}{\s@tempcntb}
873         \ifnum\s@tempcntb=\z@
874           \else
875             \reducelonercolcell{\the\s@lcnta}{\the\s@tempcntb}
876             \iflonerchanged
877               \advance\s@dncol 10\relax
878             \fi
879           \fi
880         \fi
881       \advance\s@dncol\@ne
882     \repeat
883 \egroup}
884

```

`\sumcolsets` Sum all the solution sets of the unsolved cells in the current column (`\s@lcnta`). The result is in `\sumctr`.

```

885 %% add the sets in the current column
886 \newcommand*{\sumcolsets}{\bgroup
887   \s@dnrow=\s@lcnta
888   \toprangectr=\s@dnrow
889   \advance\toprangectr 73\relax
890   \global\sumctr\z@
891   \loop
892     \ifnum\s@dnrow<\toprangectr
893     \ifnum\useknt{s@lans\the\s@dnrow}>\z@
894       \global\advance\sumctr\useknt{s@lans\the\s@dnrow}
895     \fi
896     \advance\s@dnrow 9\relax
897   \repeat
898 \egroup}
899

```

`\reducelonercolcell` `\reducelonercolcell{<colcell>}{<digit>}`. Given (the cell at) the start of a column and a lone digit, find the cell in the column with that digit in its solution set, and make the digit the solution of that cell.

```

900 %% \reducelonercolcell{startofcolcell}{digit}
901 %% given a col and a lone digit, find the cell containing the digit
902 %% and make the digit a solution of that cell.
903 \newcommand*{\reducelonercolcell}[2]{\bgroup
904   \s@dnrow=#1
905   \toprangectr=\s@dnrow
906   \advance\toprangectr 73\relax
907   \ifnum #2<\@ne
908     \advance\s@dnrow 7300\relax
909   \fi
910   \loop
911     \ifnum\s@dnrow<\toprangectr
912     \numofnuminset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\lonecellctr}
913     \ifnum\lonecellctr=\@ne%   this cell has the loner
914       \lonecellctr=#2
915       \global\useknt{s@lans\the\s@dnrow}=-\lonecellctr
916       \global\lonerchangedtrue
917       \global\advance\solcnt\@ne
918 \commentary{\the\solcnt\space solved (#2 loner in column starting #1)}%
919       \s@dnrow=\toprangectr
920     \fi
921     \advance\s@dnrow 9\relax
922   \repeat
923 \egroup}
924

```

`\reduceboxloners` Loop through the boxes and reduce the first loner found.

```

925 %% Loop through the boxes, reducing loners

```

```

926 \newcommand*{\reduceboxloners}{\bgroup
927   \sumctr\z@
928   \boxctr\@ne
929   \loop
930   \ifnum\boxctr<10\relax

```

The next bit of code calculates the corner cell of the Nth box.

```

931     \ifcase\boxctr
932     \or \s@lcnta=\@ne
933     \or \s@lcnta=4\relax
934     \or \s@lcnta=7\relax
935     \or \s@lcnta=28\relax
936     \or \s@lcnta=31\relax
937     \or \s@lcnta=34\relax
938     \or \s@lcnta=55\relax
939     \or \s@lcnta=58\relax
940     \or \s@lcnta=61\relax
941     \fi
942     \sumboxsets
943     \ifnum\sumctr>\z@
944       \getloner{\the\sumctr}{\s@dttempcntb}
945       \ifnum\s@dttempcntb=\z@
946         \else
947           \reducelonerboxcell{\the\s@lcnta}{\the\s@dttempcntb}
948           \iflonerchanged
949             \advance\boxctr 10\relax
950           \fi
951         \fi
952       \fi
953     \advance\boxctr\@ne
954   \repeat
955 \egroup}
956

```

`\sumboxsets` Sum all the solution sets of the unsolved cells in the current box (`\s@lcnta`). The result is in `\sumctr`.

```

957 %% add the sets in the current box
958 \newcommand*{\sumboxsets}{\bgroup
959   \s@dncol=\s@lcnta
960   \toprangectr=\s@dncol
961   \advance\toprangectr 3\relax
962   \global\sumctr\z@
963   \loop
964     \ifnum\s@dncol<\toprangectr
965     \s@dnrow=\s@dncol
966     \ifnum\useknt{s@lans\the\s@dnrow}>\z@
967       \global\advance\sumctr\useknt{s@lans\the\s@dnrow}
968     \fi
969     \advance\s@dnrow 9\relax
970     \ifnum\useknt{s@lans\the\s@dnrow}>\z@

```

```

971     \global\advance\sumctr\useknt{s@lans\the\s@dnrow}
972     \fi
973     \advance\s@dnrow 9\relax
974     \ifnum\useknt{s@lans\the\s@dnrow}>\z@
975         \global\advance\sumctr\useknt{s@lans\the\s@dnrow}
976     \fi
977     \advance\s@dncol\@ne
978     \repeat
979 \egroup}
980

```

`\reducelonerboxcell` `\reducelonerboxcell{(boxcell)}{(digit)}`. Given (the cell at) the start of a box and a lone digit, find the cell in the box with that digit in its solution set, and make the digit the solution of that cell.

```

981 %% \reducelonerboxcell{startofboxcell}{digit}
982 %% given a box and a lone digit, find the cell containing the digit
983 %% and make the digit a solution of that cell.
984 \newcommand*{\reducelonerboxcell}[2]{\bgroup
985     \s@dncol=\s@lcnta
986     \toprangectr=\s@dncol
987     \advance\toprangectr 3\relax
988     \global\sumctr\z@
989     \loop
990         \ifnum\s@dncol<\toprangectr
991             \s@dnrow=\s@dncol
992             \numofnuminset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\lonecellctr}
993             \ifnum\lonecellctr=\@ne%    this cell has the loner
994                 \lonecellctr=#2
995                 \global\useknt{s@lans\the\s@dnrow}=-\lonecellctr
996                 \global\lonerchangedtrue
997                 \global\advance\solcnt\@ne
998 \commentary{\the\solcnt\space solved (#2 loner in box starting #1)}%
999                 \s@dncol=\toprangectr
1000             \fi
1001         \iflonerchanged\else
1002             \advance\s@dnrow 9\relax
1003             \numofnuminset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\lonecellctr}
1004             \ifnum\lonecellctr=\@ne%    this cell has the loner
1005                 \lonecellctr=#2
1006                 \global\useknt{s@lans\the\s@dnrow}=-\lonecellctr
1007                 \global\lonerchangedtrue
1008                 \global\advance\solcnt\@ne
1009 \commentary{\the\solcnt\space solved (#2 loner in box starting #1)}%
1010             \s@dncol=\toprangectr
1011         \fi
1012     \fi
1013     \iflonerchanged\else
1014         \advance\s@dnrow 9\relax
1015         \numofnuminset{#2}{\the\useknt{s@lans\the\s@dnrow}}{\lonecellctr}
1016         \ifnum\lonecellctr=\@ne%    this cell has the loner

```

```

1017     \lonecellctr=#2
1018     \global\useknt{s@lans\the\s@dnrow}=-\lonecellctr
1019     \global\lonerchangedtrue
1020     \global\advance\solcnt\@ne
1021 \commentary{\the\solcnt\space solved (#2 loner in box starting #1)}%
1022     \s@dncol=\toprangectr
1023     \fi
1024     \fi
1025     \advance\s@dncol\@ne
1026     \repeat
1027 \egroup}
1028
1029 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% reduce pairs %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1030

```

`\reducepairs` Perform pair reduction, reducing the first pair found. First try by row, then by column, and lastly by box.

```

1031 %% reduce the pairs, per row, column and box
1032 \newcommand*{\reducepairs}{\bgroup
1033 \commentary{reducepairs}% *****}
1034 \global\pairchangedfalse
1035 \ifnum\difficultyctr<8\relax
1036 \global\difficultyctr=8\relax
1037 \fi
1038 \ifnum\solcnt>80\else
1039 \reducerowpairs
1040 \ifpairchanged\else
1041 \reducecolpairs
1042 \ifpairchanged\else
1043 \reduceboxpairs
1044 \fi
1045 \fi
1046 \fi
1047 \egroup}
1048

```

`\reducerowpairs` Loop through the rows and reduce the first pair found.

```

1049 %% Loop through the rows, reducing pairs
1050 \newcommand*{\reducerowpairs}{%
1051 \sumctr\z@
1052 \s@dnrow\@ne
1053 \loop
1054 \ifnum\s@dnrow<10\relax
1055 \s@lcnta=\s@dnrow \advance\s@lcnta\m@ne
1056 \multiply\s@lcnta 9\relax \advance\s@lcnta\@ne
1057 \sumrowsets
1058 \ifnum\sumctr>\z@
1059 \reducearowpair{\the\s@lcnta}
1060 \ifpairchanged

```

```

1061     \advance\s@dnrow 10\relax
1062     \fi
1063     \fi
1064     \advance\s@dnrow\@ne
1065 \repeat}
1066

```

`\reducearowpair` `\reducearowpair{<rowcell>}` reduces a pair in the row starting with cell `<rowcell>`.

```

1067 %% reducearowpair{row}
1068 \newcommand*{\reducearowpair}[1]{%
1069   \bgroup
1070   \s@dncol=#1\relax
1071   \toprangectr=\s@dncol
1072   \advance\toprangectr 9\relax
1073   \maxrangectr=\toprangectr
1074   \advance\maxrangectr\m@ne
1075   \global\sumctr\z@
1076   \loop
1077     \ifnum\s@dncol<\maxrangectr
1078     \asetctr=\useknt{s@lans\the\s@dncol}%
1079     \ifnum\asetctr>\z@
1080       \findrowpair
1081     \fi
1082     \advance\s@dncol\@ne
1083   \repeat
1084 \egroup}
1085

```

`\findrowpair` `\findrowpair` attempts to find a pair in the current row.

```

1086 \newcommand*{\findrowpair}{\bgroup
1087   \secondctr=\s@dncol
1088   \advance\secondctr\@ne
1089   \loop
1090     \ifnum\secondctr<\toprangectr
1091     \ifnum\asetctr=\useknt{s@lans\the\secondctr}
1092     \checksetforpair{\the\asetctr}
1093     \ifnum\numdigitsctr=\tw@
1094 %% first row cell is \s@lcnta
1095 %% first pair cell is \s@dncol, second pair cell is \secondctr
1096 %% digits are \digitictr and \digitiictr
1097     \deleterowpairedigits
1098     \fi
1099     \fi
1100     \advance\secondctr\@ne
1101   \repeat
1102 \egroup}
1103

```

`\checksetforpair` `\checksetforpair{<set>}` attempts to find a pair in a set. It sets `\digitictr` and `\digitiictr` as the first and second digits in the set and `\numdigitsctr` to

the total number of digits in the set (which is 2 for a pair).

```

1104 %% \checksetforpair{set} returns with:
1105 %%     \digitictr = first digit
1106 %%     \digitiictr = second digit
1107 %%     \numdigitsctr = number of digits (2 for a pair)
1108 \newcommand*\checksetforpair}[1]{\bgroup
1109   \s@temploopcnta\@ne
1110   \global\numdigitsctr\z@
1111   \global\digitictr\z@
1112   \global\digitiictr\z@
1113   \loop
1114     \ifnum\s@temploopcnta<10\relax
1115     \numofnuminset{\the\s@temploopcnta}{#1}{\s@tempcnta}
1116     \ifnum\s@tempcnta=\@ne
1117       \global\advance\numdigitsctr\@ne
1118       \ifcase\numdigitsctr
1119         \or \global\digitictr=\s@temploopcnta
1120         \or \global\digitiictr=\s@temploopcnta
1121         \else \advance\s@temploopcnta 10\relax
1122         \fi
1123       \fi
1124       \advance\s@temploopcnta\@ne
1125     \repeat
1126 \egroup}
1127

```

`\deleterowpairedigits` `\deleterowpairedigits` deletes the pair digits from non-pair cells in a row. The first cell in the row is `\s@lcnta`, the cells that contain the two pairs are `\s@dncol` and `\secondctr`, and the digits are `\digitictr` and `\digitiictr`.

```

1128 %% first row cell is \s@lcnta
1129 %% first pair cell is \s@dncol, second pair cell is \secondctr
1130 %% digits are \digitictr and \digitiictr
1131 \newcommand*\deleterowpairedigits{\bgroup
1132   \global\pairchangedfalse
1133   \s@temploopcnta=\s@lcnta
1134   \toprangectr=\s@temploopcnta \advance\toprangectr 9\relax
1135   \loop
1136     \ifnum\s@temploopcnta<\toprangectr
1137     \ifnum\s@temploopcnta=\s@dncol
1138     \else
1139       \ifnum\s@temploopcnta=\secondctr
1140       \else
1141         \deletenumfromset{\the\digitictr}%
1142           {\the\useknt{s@lans\the\s@temploopcnta}}{\s@tempcnta}
1143         \ifsetchanged
1144           \global\useknt{s@lans\the\s@temploopcnta}=\s@tempcnta
1145           \global\pairchangedtrue
1146         \fi
1147         \deletenumfromset{\the\digitiictr}%

```

```

1148             {\the\useknt{s@lans\the\s@dtemploopcnta}}{\s@dttempcnta}
1149         \ifsetchanged
1150             \global\useknt{s@lans\the\s@dtemploopcnta}=\s@dttempcnta
1151             \global\pairchangedtrue
1152         \fi
1153     \fi
1154 \fi
1155 \advance\s@dtemploopcnta\@ne
1156 \repeat
1157 \egroup}
1158

```

`\reducecolpairs` Loop through the rows and reduce the first pair found.

```

1159 %% Loop through the columns, reducing pairs
1160 \newcommand*{\reducecolpairs}{%
1161     \sumctr\z@
1162     \s@dncol\@ne
1163     \loop
1164         \ifnum\s@dncol<10\relax
1165             \s@lcnta=\s@dncol
1166             \sumcolsets
1167             \ifnum\sumctr>\z@
1168                 \reduceacolpair{\the\s@lcnta}
1169                 \ifpairchanged
1170                     \advance\s@dncol 10\relax
1171             \fi
1172         \fi
1173     \advance\s@dncol\@ne
1174 \repeat}
1175

```

`\reduceacolpair` `\reduceacolpair{<colcell>}` reduces a pair in the column starting with cell `<colcell>`.

```

1176 %% reduceacolpair{col}
1177 \newcommand*{\reduceacolpair}[1]{\bgroup
1178     \s@dnrow=#1\relax
1179     \toprangectr=\s@dnrow
1180     \advance\toprangectr 73\relax
1181     \maxrangectr=\toprangectr
1182     \advance\maxrangectr -9\relax
1183     \global\sumctr\z@
1184     \loop
1185         \ifnum\s@dnrow<\maxrangectr
1186             \asetctr=\useknt{s@lans\the\s@dnrow}%
1187             \ifnum\asetctr>\z@
1188                 \findcolpair
1189             \fi
1190         \advance\s@dnrow 9\relax
1191     \repeat
1192 \egroup}

```

1193

`\findcolpair` `\findcolpair` attempts to find a pair in the current column.

```

1194 \newcommand*{\findcolpair}{\bgroup
1195   \secondctr=\s@dnrow
1196   \advance\secondctr 9\relax
1197   \loop
1198     \ifnum\secondctr<\toprangectr
1199       \ifnum\asetctr=\useknt{s@lans\the\secondctr}
1200         \checksetforpair{\the\asetctr}
1201         \ifnum\numdigitsctr=\tw@
1202 %% first row cell is \s@lcnta
1203 %% first pair cell is \s@dnrow, second pair cell is \secondctr
1204 %% digits are \digitictr and \digitiictr
1205         \deletecolpairdigits
1206         \fi
1207       \fi
1208       \advance\secondctr 9\relax
1209     \repeat
1210 \egroup}
1211

```

`\deletecolpairdigits` `\deletecolpairdigits` deletes the pair digits from non-pair cells in a column. The first cell in the column is `\s@lcnta`, the cells that contain the two pairs are `\s@dnrow` and `\secondctr`, and the digits are `\digitictr` and `\digitiictr`.

```

1212 %% first col cell is \s@lcnta
1213 %% first pair cell is \s@dnrow, second pair cell is \secondctr
1214 %% digits are \digitictr and \digitiictr
1215 \newcommand*{\deletecolpairdigits}{\bgroup
1216   \global\pairchangedfalse
1217   \s@dtemploopcnta=\s@lcnta
1218   \toprangectr=\s@dtemploopcnta \advance\toprangectr 73\relax
1219   \loop
1220     \ifnum\s@dtemploopcnta<\toprangectr
1221     \ifnum\s@dtemploopcnta=\s@dnrow
1222     \else
1223       \ifnum\s@dtemploopcnta=\secondctr
1224       \else
1225         \deletenumfromset{\the\digitictr}%
1226           {\the\useknt{s@lans\the\s@dtemploopcnta}}{\s@dtempcnta}
1227         \ifsetchanged
1228           \global\useknt{s@lans\the\s@dtemploopcnta}=\s@dtempcnta
1229           \global\pairchangedtrue
1230         \fi
1231         \deletenumfromset{\the\digitiictr}%
1232           {\the\useknt{s@lans\the\s@dtemploopcnta}}{\s@dtempcnta}
1233         \ifsetchanged
1234           \global\useknt{s@lans\the\s@dtemploopcnta}=\s@dtempcnta
1235         \global\pairchangedtrue

```

```

1236     \fi
1237     \fi
1238     \fi
1239     \advance\s@dtemploopcnta 9\relax
1240     \repeat
1241 \egroup}
1242

```

`\reduceboxpairs` Loop through the boxes and reduce the first pair found.

```

1243 %% Loop through the boxes, reducing pairs
1244 \newcommand*\reduceboxpairs{\bgroup
1245   \sumctr\z@
1246   \boxctr\@ne
1247   \loop
1248     \ifnum\boxctr<10\relax
1249     \ifcase\boxctr
1250     \or \s@lcnta=\@ne
1251     \or \s@lcnta=4\relax
1252     \or \s@lcnta=7\relax
1253     \or \s@lcnta=28\relax
1254     \or \s@lcnta=31\relax
1255     \or \s@lcnta=34\relax
1256     \or \s@lcnta=55\relax
1257     \or \s@lcnta=58\relax
1258     \or \s@lcnta=61\relax
1259     \fi
1260     \sumboxsets
1261     \ifnum\sumctr>\z@
1262     \reduceaboxpair{the\s@lcnta}
1263     \ifpairchanged
1264     \advance\boxctr 10\relax
1265     \fi
1266     \fi
1267     \advance\boxctr\@ne
1268   \repeat
1269 \egroup}
1270

```

`\getnthboxcell` `\getnthboxcell{<n>}{<firstcell>}{<cnt>}`. Given the first cell (`<firstcell>`) in a box and a number, `<n>`, between 1 and 9, set the `\count` `<cnt>` to the cell number of the `<n>`th cell in the box.

```

1271 %% \getnthboxcell{n}{firstcell}{returnctr}
1272 \newcommand*\getnthboxcell[3]{\bgroup
1273   \s@dtempcnta=#2
1274   \ifcase#1
1275   \or
1276   \or \advance\s@dtempcnta 9\relax
1277   \or \advance\s@dtempcnta 18\relax
1278   \or \advance\s@dtempcnta \@ne

```

```

1279 \or \advance\s@dttempcnta 10\relax
1280 \or \advance\s@dttempcnta 19\relax
1281 \or \advance\s@dttempcnta \tw@
1282 \or \advance\s@dttempcnta 11\relax
1283 \or \advance\s@dttempcnta 20\relax
1284 \fi
1285 \global #3=\s@dttempcnta
1286 \egroup}
1287

```

`\reduceboxpair` `\reduceboxpair{<boxcell>}` reduces a pair in the box starting with cell `<boxcell>`.

```

1288 %% \reduceboxpair{startofboxcell}
1289 \newcommand*{\reduceboxpair}[1]{\bgroup
1290 \s@dttemploopcnta@one
1291 \toprangectr=\s@dttemploopcnta
1292 \advance\toprangectr 9\relax
1293 \maxrangectr=\toprangectr
1294 \advance\maxrangectr\m@ne
1295 \loop
1296 \ifnum\s@dttemploopcnta<\maxrangectr
1297 \getnthboxcell{\the\s@dttemploopcnta}{#1}{\s@dttempcnta}
1298 \asetctr=\useknt{s@lans\the\s@dttempcnta}
1299 \ifnum\asetctr>\z@
1300 \s@dnrow=\s@dttemploopcnta
1301 \findboxpair{#1}
1302 \fi
1303 \advance\s@dttemploopcnta\@ne
1304 \repeat
1305 \egroup}
1306

```

`\findboxpair` `\findboxpair` attempts to find a pair in the current box, and then does a pair reduction for the box.

```

1307 %% \findboxpair{startofboxcell}
1308 \newcommand*{\findboxpair}[1]{%
1309 \bgroup
1310 \secondctr=\s@dttemploopcnta
1311 \advance\secondctr\@ne
1312 \loop
1313 \ifnum\secondctr<\toprangectr
1314 \getnthboxcell{\the\secondctr}{#1}{\s@dttempcntb}
1315 \ifnum\asetctr=\useknt{s@lans\the\s@dttempcntb}
1316 \checksetforpair{\the\asetctr}
1317 \ifnum\numdigitsctr=\tw@
1318 %% first box cell is #1 (=s@lcnta)
1319 %% first pair cell is \s@dnrow, second pair cell is \secondctr
1320 %% digits are \digitictr and \digitiictr
1321 \deleteboxpairedigits{#1}
1322 \fi
1323 \fi

```

```

1324     \advance\secondctr \@ne
1325     \repeat
1326 \egroup}
1327

\deleteboxpairdigits \deleteboxpairdigits{⟨boxcell⟩} deletes the pair digits from non-pair cells in a
box. The first cell in the box is ⟨boxcell⟩, the cells that contain the two pairs are
\s@dnrow and \secondctr, and the digits are \digitictr and \digitiictr.

1328 %% first box cell is #1
1329 %% first pair cell is \s@dnrow, second pair cell is \secondctr
1330 %% digits are \digitictr and \digitiictr
1331 \newcommand*{\deleteboxpairdigits}[1]{\bgroup
1332   \global\pairchangedfalse
1333   \s@dttemploopcnta=\@ne
1334   \toprangelctr=\s@dttemploopcnta \advance\toprangelctr 9\relax
1335   \loop
1336     \ifnum\s@dttemploopcnta<\toprangelctr
1337     \ifnum\s@dttemploopcnta=\s@dnrow
1338     \else
1339       \ifnum\s@dttemploopcnta=\secondctr
1340       \else
1341         \getnthboxcell{\the\s@dttemploopcnta}{#1}{\s@dttempcntb}
1342         \deletenumfromset{\the\digitictr}%
1343           {\the\useknt{s@lans\the\s@dttempcntb}}{\s@dttempcnta}
1344         \ifsetchanged
1345           \global\useknt{s@lans\the\s@dttempcntb}=\s@dttempcnta
1346           \global\pairchangedtrue
1347         \fi
1348         \deletenumfromset{\the\digitiictr}%
1349           {\the\useknt{s@lans\the\s@dttempcntb}}{\s@dttempcnta}
1350         \ifsetchanged
1351           \global\useknt{s@lans\the\s@dttempcntb}=\s@dttempcnta
1352           \global\pairchangedtrue
1353         \fi
1354       \fi
1355     \fi
1356     \advance\s@dttemploopcnta \@ne
1357   \repeat
1358 \egroup}
1359

```

3.3.1 Checking the solution

```

1360 %%%%%%%%%%%%%%% check solution %%%%%%%%%%%%%%%
1361

```

We have found a solution when all the binary sets are negative. It's less efficient, but probably more useful, to check each row, column, and block (3 times 81 tests instead of 81).

`\checksolution` Check the solution by rows, columns, and boxes.

```
1362 \newcommand*{\checksolution}{%
1363   \checkrows
1364   \checkcols
1365   \checkboxes}
1366
```

`\checkrows` Check the row solutions. A row is said to be correct if a solution has been assigned to each cell in the row, thus relying on the assumption that the solver never makes a mistake. The check is not sufficient for a solution obtained in another manner.

```
1367 \newcommand*{\checkrows}{%
1368   \sumctr\z@
1369   \s@dnrow\@ne
1370   \loop
1371     \ifnum\s@dnrow<10\relax
1372       \s@lcnta=\s@dnrow \advance\s@lcnta\m@ne
1373       \multiply\s@lcnta 9\relax \advance\s@lcnta\@ne
1374       \sumrowsets
1375       \ifnum\sumctr=\z@\else
1376 \typeout{Row \the\s@dnrow\space is incorrect}%
1377         Row \the\s@dnrow\space is incorrect. \\
1378         \fi
1379       \advance\s@dnrow\@ne
1380     \repeat}
1381
```

`\checkcols` Check the column solutions.

```
1382 \newcommand*{\checkcols}{%
1383 \bgroup
1384   \sumctr\z@
1385   \s@dncol\@ne
1386   \loop
1387     \ifnum\s@dncol<10\relax
1388       \s@lcnta=\s@dncol
1389       \sumcolsets
1390       \ifnum\sumctr=\z@\else
1391 \typeout{Column \the\s@dncol\space is incorrect}%
1392         Column \the\s@dncol\space is incorrect. \\
1393         \fi
1394       \advance\s@dncol\@ne
1395     \repeat
1396 \egroup}
1397
```

`\checkboxes` Check the box solutions.

```
1398 \newcommand*{\checkboxes}{%
1399 \bgroup
1400   \sumctr\z@
1401   \boxctr\@ne
```

```

1402 \loop
1403   \ifnum\boxctr<10\relax
1404 \ifcase\boxctr
1405 \or \s@lcnta=\@ne
1406 \or \s@lcnta=4\relax
1407 \or \s@lcnta=7\relax
1408 \or \s@lcnta=28\relax
1409 \or \s@lcnta=31\relax
1410 \or \s@lcnta=34\relax
1411 \or \s@lcnta=55\relax
1412 \or \s@lcnta=58\relax
1413 \or \s@lcnta=61\relax
1414 \fi
1415   \sumboxsets
1416   \ifnum\sumctr=\z@\else
1417 \typeout{Box \the\boxctr\space is incorrect}%
1418   Box \the\boxctr\space is incorrect. \\
1419   \fi
1420   \advance\boxctr\@ne
1421 \repeat
1422 \egroup}
1423

```

`\typesimplestatusX` May be used to display the current status of the (partial) solution following simple reductions.

```

1424 %% typeset the current status of the partial solution
1425 \newcommand*\typesimplestatusX}{%
1426 \par\noindent\begin{minipage}{\linewidth}
1427 \begin{center}
1428 Simple reductions stopped (with \the\solcnt\space solutions)\!
1429 \footnotesize
1430 \displaystatus
1431 \end{center}
1432 \end{minipage}}
1433

```

`\typelonestatusX` May be used to display the current status of the (partial) solution after a loner reduction.

```

1434 %% typeset the current status after loner processing
1435 \newcommand*\typelonestatusX}{%
1436 \par\noindent\begin{minipage}{\linewidth}
1437 \begin{center}
1438 Loners processed (with \the\solcnt\space solutions)\!
1439 \footnotesize
1440 \displaystatus
1441 \end{center}
1442 \end{minipage}}
1443

```

`\hideprogress` Normally the progress towards a solution is not displayed. It will be displayed in `\showprogress`

the document, though, after the `\showprogress` declaration. The `\hideprogress` declaration switches off the displays.

```

1444 %% For curiosity as to how a solution is developing call the \showprogress
1445 %% declaration (\hideprograss turns this off).
1446 \newcommand*{\hideprogress}{%
1447   \let\typesimplestatus\relax
1448   \let\typelonestatus\relax}
1449 \newcommand*{\showprogress}{%
1450   \let\typesimplestatus\typesimplestatusX
1451   \let\typelonestatus\typelonestatusX}
1452 \hideprogress
1453 %\showprogress
1454
1455 %%%%%%%%%%% write solution to a game file %%%%%%%%%%%
1456

```

`\sudsolnfile` `\writegame` writes the (partial) solution to the file `\sudsolnfile` (default `\writegame` `sud.out`).

```

1457 %% write out the solution to game file \sudsolnfile.
1458 \newcommand*{\sudsolnfile}{sud.out}
1459 \newcommand*{\writegame}{%
1460   \immediate\closeout\s@dwrite
1461   \immediate\openout\s@dwrite=\sudsolnfile
1462   \s@dnrow\@ne
1463   \loop
1464     \ifnum\s@dnrow<81\relax
1465     \gatherline{\the\s@dnrow}
1466     \immediate\write\s@dwrite{\sudaline}
1467     \advance\s@dnrow 9\relax
1468   \repeat
1469   \immediate\closeout\s@dwrite}
1470

```

`\gatherline` `\gatherline{\rowstartcell}` collects the solutions from one row in the grid and `\sudaline` converts them into the corresponding characters for one line in a sudoku file, putting the characters into the `\sudaline` macro.

```

1471 %% \gatherline{colstart}
1472 %% collect all the characters for one line in a game file
1473 \newcommand*{\gatherline}[1]{\bgroup
1474   \gdef\sudaline{}
1475   \s@dncol=#1\relax
1476   \toprangectr=\s@dncol
1477   \advance\toprangectr 9\relax
1478   \loop
1479     \ifnum\s@dncol<\toprangectr
1480     \s@dttempcnta=\useknt{s@lans\the\s@dncol}
1481     \ifnum\s@dttempcnta>\m@ne
1482       \g@addto@macro{\sudaline}{.}

```

```

1483 \else
1484   \s@dttempcnta--\s@dttempcnta
1485   \ifcase\s@dttempcnta
1486   \or \g@addto@macro{\sudaline}{1}
1487   \or \g@addto@macro{\sudaline}{2}
1488   \or \g@addto@macro{\sudaline}{3}
1489   \or \g@addto@macro{\sudaline}{4}
1490   \or \g@addto@macro{\sudaline}{5}
1491   \or \g@addto@macro{\sudaline}{6}
1492   \or \g@addto@macro{\sudaline}{7}
1493   \or \g@addto@macro{\sudaline}{8}
1494   \or \g@addto@macro{\sudaline}{9}
1495   \else \g@addto@macro{\sudaline}{.}
1496   \fi
1497 \fi
1498 \advance\s@dncol\@ne
1499 \repeat
1500 \egroup}
1501
1502 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% THE END GOAL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1503 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% \sudoksolve{<file>} %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1504

```

`\sudokusolve` `\sudoksolve{<file>}` reads a sudoku puzzle from the *<file>* sudoku file, typesets the problem, does its best to solve it. It writes the solution to the `\sudsolnfile` file, then reads it back in again and typesets it.

```

1505 %% Read in and typeset the problem, solve it, and typeset the answer
1506 \newcommand{\sudokusolve}[1]{%
1507   \begin{center}
1508     \sudoku{#1}%
1509   \end{center}
1510   \getproblem{#1}%
1511   \reduceallcells
1512   \keepreducing
1513   \writegame
1514   \noindent\begin{minipage}{\linewidth}
1515     THE ANSWER
1516     \begin{center}
1517       \cluefont\normalsize
1518       \cellsize{1.5\baselineskip}
1519       \sudoku{\sudsolnfile}
1520     \end{center}
1521   \end{minipage}}
1522

```

The end of the package.

```
1523 </solve>
```

3.4 The createsudoku package

The createsudoku package lets you automatically generate Sudoku puzzles. The package requires Donald Arseneau's `random.tex` and also the solvesudoku package, which in turn requires the printsudoku package.

```

1524 <*create>
1525 \NeedsTeXFormat{LaTeX2e}
1526 \ProvidesPackage{createsudoku}[2006/02/16 v1.0 create sudoku puzzles]
1527 \input{random}
1528 \RequirePackage{solvesudoku}
1529

```

`\oldcommentary` To avoid clutter it's a good idea to switch off solvesudoku's commentary. On the other hand it could be useful for this package to provide its own commentary.

```

\commentary
\gencommentary 1530 \let\oldcommentary\commentary
1531 \let\commentary@gobble
1532 \newcommand*{\gencommentary}[1]{\typeout{#1}}
1533

```

`\genfile` We need a file to write out the puzzle to, and we also need two temporary files.

```

\prevfile 1534 \newcommand*{\genfile}{gensud.sud}
\currfile 1535 \newcommand*{\prevfile}{genprev.sdx}
1536 \newcommand*{\currfile}{gencurr.sdx}
1537

```

`\randomi` This sets the seed for the random number generator. The default is setting it to `\setsudrandom -1`, which makes the seed depend on the date and time.

```

1538 \randomi@m@ne
1539 \newcommand*{\setsudrandom}[1]{%
1540 \randomi=#1\relax
1541 \ifnum\randomi<\@ne\else
1542 \nextrand \nextrand \nextrand
1543 \typeout{random initialised to #1}%
1544 \fi}
1545

```

`\generategrid` `\generategrid[file]` generates a new puzzle. If the sudoku file *file* is specified this is taken for the starting grid, otherwise a default one is used. The starting grid should be a complete solution.

The pairs of rows, and pairs of columns, in the starting grid are exchanged in a random fashion (`\swaps`) and then the cell solutions are eliminated (`\elimclues`) until the puzzle is sufficiently complicated.

```

1546 \newcommand*{\generategrid}[1][\@empty]{%
1547 \ifx\@empty #1
1548 \writestartgrid
1549 \getproblem{\prevfile}
1550 \else
1551 \getproblem{#1}

```

```

1552 \fi
1553 \swaps
1554 \elimclues
1555 \printsudresults}
1556 \newcommand*{\printsudresults}{%
1557 \sudokusolve{\genfile}%
1558 }
1559

```

`\writestartgrid` Write out an initial fully solved grid to `\prevfile`.

```

1560 %% the starting data
1561 \newcommand*{\writestartgrid}{%
1562 \immediate\closeout\s@dwrite
1563 \immediate\openout\s@dwrite=\prevfile
1564 \immediate\write\s@dwrite{123456789}
1565 \immediate\write\s@dwrite{456789123}
1566 \immediate\write\s@dwrite{789123456}
1567 \immediate\write\s@dwrite{234567891}
1568 \immediate\write\s@dwrite{567891234}
1569 \immediate\write\s@dwrite{891234567}
1570 \immediate\write\s@dwrite{345678912}
1571 \immediate\write\s@dwrite{678912345}
1572 \immediate\write\s@dwrite{912345678}
1573 \immediate\closeout\s@dwrite}
1574

```

`\swaps` `\swaps` randomly exchanges the contents of pairs of columns and pairs of rows. In order to preserve the number uniqueness constraint for boxes, only pairs of columns/rows that are in the same box may be exchanged.

```

1575 \newcommand*{\swaps}{\bgroup
1576 \setrannum{\maxrangectr}{36}{48}%
1577 \gencommentary{number of swaps (36--48) = maxrangectr = \the\maxrangectr}
1578 \s@dtemploopcnta\@ne
1579 \loop
1580 \ifnum\s@dtemploopcnta<\maxrangectr
1581 \setrannum{\s@dtempcntb}{1}{18}%
1582 %%\typeout{\space\space\space swap no \the\s@dtemploopcnta\space is \the\s@dtempcntb}
1583 \ifcase\s@dtempcntb
1584 \or \swapcolpair{1}{2}%
1585 \or \swapcolpair{1}{3}%
1586 \or \swapcolpair{2}{3}%
1587 \or \swapcolpair{4}{5}%
1588 \or \swapcolpair{4}{6}%
1589 \or \swapcolpair{5}{6}%
1590 \or \swapcolpair{7}{8}%
1591 \or \swapcolpair{7}{9}%
1592 \or \swapcolpair{8}{9}%
1593 \or \swaprowpair{1}{10}%
1594 \or \swaprowpair{1}{19}%

```

```

1595 \or \swaprowpair{10}{19}%
1596 \or \swaprowpair{28}{37}%
1597 \or \swaprowpair{28}{46}%
1598 \or \swaprowpair{37}{46}%
1599 \or \swaprowpair{55}{64}%
1600 \or \swaprowpair{55}{73}%
1601 \or \swaprowpair{64}{73}%
1602 \fi
1603 \advance\s@dtemploopcnta\@ne
1604 \repeat
1605 \egroup}
1606

```

`\swaprowpair` `\swaprowpair{⟨rowstartcell1⟩}{⟨rowstartcell2⟩}` swaps the contents of the two rows starting at `⟨rowstartcell1⟩` and `⟨rowstartcell2⟩`.

```

1607 %% swap the data in rows starting at #1, and #2
1608 \newcommand*\swaprowpair}[2]{\bgroup
1609 \s@dnrow=#1\relax
1610 \s@drowpos=#2\relax
1611 \s@dtemploopcnta\@ne
1612 \maxrangectr=10\relax
1613 \loop
1614 \ifnum\s@dtemploopcnta<\maxrangectr
1615 \s@dttempcnta=\useknt{s@lans\the\s@dnrow}%
1616 \global\useknt{s@lans\the\s@dnrow}=\useknt{s@lans\the\s@drowpos}%
1617 \global\useknt{s@lans\the\s@drowpos}=\s@dttempcnta
1618 \advance\s@dnrow\@ne
1619 \advance\s@drowpos\@ne
1620 \advance\s@dtemploopcnta\@ne
1621 \repeat
1622 \egroup}
1623

```

`\swapcolpair` `\swapcolpair{⟨colstartcell1⟩}{⟨colstartcell2⟩}` swaps the contents of the two columns starting at `⟨colstartcell1⟩` and `⟨colstartcell2⟩`.

```

1624 %% swap the data in cols starting #1 and #2
1625 \newcommand*\swapcolpair}[2]{\bgroup
1626 \s@dnrow=#1\relax
1627 \s@drowpos=#2\relax
1628 \s@dtemploopcnta\@ne
1629 \maxrangectr=10\relax
1630 \loop
1631 \ifnum\s@dtemploopcnta<\maxrangectr
1632 \s@dttempcnta=\useknt{s@lans\the\s@dnrow}%
1633 \global\useknt{s@lans\the\s@dnrow}=\useknt{s@lans\the\s@drowpos}%
1634 \global\useknt{s@lans\the\s@drowpos}=\s@dttempcnta
1635 \advance\s@dnrow 9\relax
1636 \advance\s@drowpos 9\relax
1637 \advance\s@dtemploopcnta\@ne

```

```

1638 \repeat
1639 \egroup}
1640

\elimclues \elimclues first eliminates a set of clues from the (full) grid, writing the resulting
puzzle to the two temporary files \prevfile and \currfile. It then eliminates
clues one-by-one until the puzzle has been finally generated, the result having been
written to the \prevfile, then writing it out to the file \genfile.

1641 \newcommand*{\elimclues}{\bgroup
1642 \initialelimination
1643 \writestate{\prevfile}%
1644 \writestate{\currfile}%
1645 \elimcluesonebyone
1646 \getproblem{\prevfile}
1647 \writestate{\genfile}
1648 \egroup}
1649

\elimseventeen \elimseventeen randomly eliminates the solutions for 17 cells in the grid (appar-
ently if two numbers are completely eliminated as clues in a puzzle, then at best
the solution is ambiguous).

1650 \newcommand*{\elimseventeen}{\bgroup
1651 \s@dtemploopcnta\@ne
1652 \maxrangectr 18\relax
1653 \loop
1654 \ifnum\s@dtemploopcnta<\maxrangectr
1655 \setrannum{\s@dtempcnta}{1}{81}
1656 \global\useknt{\s@lans\the\s@dtempcnta}=11111111\relax
1657 \advance\s@dtemploopcnta\@ne
1658 \repeat
1659 \egroup}
1660

\elimcross Eliminate all solutions in column 5 and row 5.

1661 %% eliminates all clues in column 5 and row 5
1662 \newcommand*{\elimcross}{\bgroup
1663 \s@dtemploopcnta 5\relax
1664 \loop
1665 \ifnum\s@dtemploopcnta<78\relax% 78 = 5 + 73
1666 \global\useknt{\s@lans\the\s@dtemploopcnta}=11111111\relax
1667 \advance\s@dtemploopcnta 9\relax
1668 \repeat
1669 \s@dtemploopcnta 37\relax
1670 \loop
1671 \ifnum\s@dtemploopcnta<46\relax% 46 = 37 + 9
1672 \global\useknt{\s@lans\the\s@dtemploopcnta}=11111111\relax
1673 \advance\s@dtemploopcnta \@ne
1674 \repeat
1675 \egroup}

```

1676

`\elimnum` `\elimnum{⟨num⟩}` ($0 < \text{num} < 10$) eliminates all instances of the `⟨num⟩` solution from the grid.

```
1677 \newcommand*{\elimnum}[1]{\bgroup
1678   \s@dttemploopcnta\@ne
1679   \loop
1680     \ifnum\s@dttemploopcnta<82\relax
1681     \ifnum\useknt{s@lans\the\s@dttemploopcnta}=#1\relax
1682       \global\useknt{s@lans\the\s@dttemploopcnta}=11111111\relax
1683     \fi
1684     \advance\s@dttemploopcnta \@ne
1685   \repeat
1686 \egroup}
1687
```

`\elimcrossandnines` Eliminate all 9's and the solutions in column 5 and row 5.

```
1688 \newcommand*{\elimcrossandnines}{\bgroup
1689   \elimcross
1690   \s@dttemploopcnta\@ne
1691   \loop
1692     \ifnum\s@dttemploopcnta<82\relax
1693     \ifnum\useknt{s@lans\the\s@dttemploopcnta}=-9\relax
1694       \global\useknt{s@lans\the\s@dttemploopcnta}=11111111\relax
1695     \fi
1696     \advance\s@dttemploopcnta \@ne
1697   \repeat
1698 \egroup}
1699
```

`\elimex` Eliminate all solutions on the diagonals of the grid.

```
1700 \newcommand*{\elimex}{\bgroup
1701   \s@dttemploopcnta\@ne
1702   \loop
1703     \ifnum\s@dttemploopcnta<82\relax
1704     \global\useknt{s@lans\the\s@dttemploopcnta}=11111111\relax
1705     \advance\s@dttemploopcnta 10\relax
1706   \repeat
1707   \s@dttemploopcnta=9\relax
1708   \loop
1709     \ifnum\s@dttemploopcnta<81\relax
1710     \global\useknt{s@lans\the\s@dttemploopcnta}=11111111\relax
1711     \advance\s@dttemploopcnta 8\relax
1712   \repeat
1713 \egroup}
1714
```

`\elimcrossandex` Eliminate all solutions in column 5 and row 5 and on the diagonals.

```
1715 \newcommand*{\elimcrossandex}{\bgroup
```

```

1716 \elimcross
1717 \elimex
1718 \egroup}
1719

```

`\elimcrossandexandnines` Eliminate all solutions in column 5 and row 5 and on the diagonals, and all 9's.

```

1720 \newcommand*{\elimcrossandexandnines}{\bgroup
1721 \elimcross
1722 \elimex
1723 \elimnum{9}
1724 \egroup}
1725

```

`\initialelimination` `\initialelimination` eliminates an initial set of solutions from the grid. Several predefined sets are provided.

```

1726 \let\initialelimination\elimseventeen
1727 \let\initialelimination\elimcross
1728 \let\initialelimination\elimcrossandnines
1729 \let\initialelimination\elimcrossandex
1730 \let\initialelimination\elimcrossandexandnines
1731

```

`\toomanyloops` Eliminate solutions from the grid one-by-one until the puzzle cannot be solved.
`\elimcluesonebyone` Then reinstate the last solution deleted to produce the final puzzle. I'm not sure if this is needed, but just in case the process stops after 81 attempts to remove solutions (highly unlikely but the random number generator could keep producing numbers corresponding to clueless cells).

We use the `\currfile` and `\prevfile` to hold the current state of affairs and the previous state.

```

1732 \newcount\toomanyloops
1733 \newcommand*{\elimcluesonebyone}{\bgroup
1734 %%\typeout{***elimcluesonebyone}
1735 \toomanyloops\z@
1736 \s@dttempcntb=81\relax
1737 \loop
1738 \ifnum\s@dttempcntb>80\relax
1739 \advance\toomanyloops\@ne

```

Read the current puzzle from `\currfile` and try to solve it.

```

1740 \getproblem{\currfile}
1741 \trysolution
1742 \ifnum\solcnt=81\relax % can delete another clue

```

The puzzle was solvable so try deleting another clue. Read the (unsolved) puzzle again from `\currfile` and write its state to `\prevfile`.

```

1743 \getproblem{\currfile}
1744 \writestate{\prevfile}

```

Now delete a clue, writing the resultant puzzle to `\currfile`, and go round the loop again.

```

1745     \deleteaclue
1746     \writestate{\currfile}%
1747     \s@tempcntb=81\relax
1748     \else % done

```

The puzzle was not solvable, and the state before the last clue deletion is in file `\prevfile`.

```

1749     \typeout{Solver failed, solcnt = \the\solcnt}
1750     \s@tempcntb=\@ne
1751     \fi
1752     \ifnum\toomanyloops>80\relax
1753     \typeout{Too much looping}
1754     \s@tempcntb=\@ne
1755     \fi
1756     \repeat
1757 \egroup}
1758

```

`\ifacluenotdeleted` Keep calling the random number generator until it returns a cell that has a solution, then delete the solution.

```

\deleteaclue
1759 \newif\ifacluenotdeleted
1760 \newcommand*\deleteaclue{\bgroup
1761   \acluenotdeletedtrue
1762   \loop
1763     \ifacluenotdeleted
1764     \setranum{\s@tempcnta}{1}{81}%
1765     \ifnum\useknt{\s@lans\the\s@tempcnta}<\z@% found a clue
1766     \gencommentary{Deleting clue for cell \the\s@tempcnta}%
1767     \global\useknt{\s@lans\the\s@tempcnta}=11111111\relax
1768     \acluenotdeletedfalse
1769     \fi
1770   \repeat
1771 \egroup}
1772

```

`\trysolution` `\trysolution` attempts to solve the current puzzle state.

```

1773 \newcommand*\trysolution{\bgroup
1774   \reduceallcells
1775   \keepreducing
1776 \egroup}
1777

```

`\writestate` `\writestate{<file>}` writes the current state (clues and blanks) to the sudoku file `<file>`.

```

1778 %% \writestate{<outputfile>}
1779 \newcommand*\writestate[1]{\bgroup
1780   \immediate\closeout\s@dwrite
1781   \immediate\openout\s@dwrite=#1
1782   \s@dnrow\@ne
1783   \loop

```

```

1784 \ifnum\s@dnrow<81\relax
1785 \gatherline{\the\s@dnrow}%
1786 \immediate\write\s@dwrite{\sudaline}%
1787 \advance\s@dnrow 9\relax
1788 \repeat
1789 \immediate\closeout\s@dwrite
1790 \egroup}
1791

```

The end of the package.

```
1792 </create>
```

4 Example sudoku puzzles

LaTeXing the following file will provide you with a selection of Sudoku puzzles. You can try solving them yourself and/or get Latex to do it for you. The `solvesudoku` package is able to solve most, but not all of the puzzles.

```

1793 (*examples)
1794 \documentclass[twoside]{article}
1795 \usepackage{comment}
1796 \usepackage{url}
1797 \usepackage{printsudoku}
1798 \newcommand*{\Lpack}[1]{\textsf{#1}}
1799 \title{Example Sudoku Puzzles}
1800 \author{Peter Wilson}
1801 \date{16 February 2006}
1802 \raggedbottom
1803 \begin{document}
1804 \maketitle
1805
1806 A Sudoku puzzle consists of a 9 by 9 array of squares, some of which have
1807 numbers in them, the numbers ranging from 1 through 9. The problem is to
1808 fill in all the squares such that each row, column, and 3 by 3 box includes
1809 the numbers 1 through 9 with no repetitions.
1810
1811 This is a collection of some 54 Sudoku puzzles which I have culled
1812 from various places. The sources include, among others:
1813 the \emph{Sudoku Solver}~\cite{SSBL} website (labelled SE, SM, SH and SVH),
1814 \emph{The Daily Telegraph}~\cite{DT2} (labelled TG, TM, and TT),
1815 \emph{The Seattle Times} (labelled ST),
1816 and some automatically generated by the \Lpack{createsudoku}
1817 package~\cite{SUDOKUBUNDLE} (labelled CS).
1818
1819 You can try solving them yourself and/or get
1820 LaTeX to do it for
1821 you. The \Lpack{solvesudoku} package~\cite{SUDOKUBUNDLE} is able to solve
1822 most, but not all of the puzzles. Other LaTeX packages may do better or worse.
1823

```

1824 The puzzles are presented in approximately increasing order of difficulty.
 1825 Each source, though, has its own set of categories which don't necessarily
 1826 agree with each other (and sometimes not within one source). I have included
 1827 an indication of the difficulty as stated by the relevant source.

1828

1829 The parameters for the puzzles created via `\Lpack{createsudoku}` are given in
 1830 the table.

1831

1832 `\begin{table}[hbp]`

1833 `\centering`

1834 `\begin{tabular}{llll} \hline`

1835 `Number & Seed & \verb?initialelimination? \\ \hline`

1836 `CS1 & 1505263462 & \verb?\elimcrossandexandnines? \\`

1837 `CS2 & 1495929734 & \verb?\elimcrossandex? \\`

1838 `CS3 & 1487762722 & \verb?\elimcrossandexandnines? \\`

1839 `CS4 & 1508763610 & \verb?\elimcrossandex? \\`

1840 `CS5 & 839068723 & \verb?\elimseventeen? \\`

1841 `CS6 & 833235143 & \verb?\elimcrossandexandnines? \\ \hline`

1842 `\end{tabular}`

1843 `\end{table}`

1844

1845 `\cleardoublepage`

1846

1847 `\cluefont{\LARGE}`

1848 `\cellsize{2.25\baselineskip}`

1849

1850 `\markboth{EASY}{EASY}`

1851 `\pagestyle{headings}`

1852

1853 `\renewcommand*{\puzzlefile}{se5.sud}`

1854 `\writepuzzle%`

1855 `{.2...5863}{56.2.3.9.}{.3...7251}%`

1856 `{.975....}{.6..47.9}{.7..286..}%`

1857 `{6.58...7.}{8....1..6}{3.7.6..4.}%`

1858 `[SSBL easy 5]`

1859 `\vfill`

1860 `\begin{minipage}{0.95\linewidth}\begin{center}`

1861 `SE5 (easy) \\`

1862 `\sudoku{se5.sud}`

1863 `\end{center}\end{minipage}`

1864

1865 `\renewcommand*{\puzzlefile}{se10.sud}`

1866 `\writepuzzle%`

1867 `{.6.497..}{.82.1.6..}{79..8.145}%`

1868 `{649.5.27.}{.7.6..5.}{.3..72.96}%`

1869 `{.2...81.}{.7..28...}{.....5.7}%`

1870 `[SSBL easy 10]`

1871 `\vfill`

1872 `\begin{minipage}{0.95\linewidth}\begin{center}`

1873 `SE10 (easy) \\`

```

1874 \sudoku{se10.sud}
1875 \end{center}\end{minipage}
1876
1877 \renewcommand*{\puzzlefile}{se15.sud}
1878 \writepuzzle%
1879 {4.67.39.2}{..16.84..}{.7...4..1}%
1880 {5...4.21.}{.2.....6.}{.68.7...9}%
1881 {6..5...9.}{..24.93..}{9.53.17.4}%
1882 [SSBL easy 15]
1883 \vfill
1884 \begin{minipage}{0.95\linewidth}\begin{center}
1885 SE15 (easy) \\
1886 \sudoku{se15.sud}
1887 \end{center}\end{minipage}
1888
1889 \renewcommand*{\puzzlefile}{se20.sud}
1890 \writepuzzle%
1891 {36.9..5..}{..54.8.2.}{.....78.9}%
1892 {2...3.68}{.83.9....}{.4.7..2.3}%
1893 {6.71..3..}{13.8.59.2}{..9.7..56}%
1894 [SSBL easy 20]
1895 \vfill
1896 \begin{minipage}{0.95\linewidth}\begin{center}
1897 SE20 (easy) \\
1898 \sudoku{se20.sud}
1899 \end{center}\end{minipage}
1900
1901 \renewcommand*{\puzzlefile}{se25.sud}
1902 \writepuzzle%
1903 {...63.5.}{6.5.1.839}{.1...2..}%
1904 {823...7..}{.5...7...}{.64.9...1}%
1905 {4.25796.3}{.9.1.65.8}{.1...89.7}%
1906 [SSBL easy 25]
1907 \vfill
1908 \begin{minipage}{0.95\linewidth}\begin{center}
1909 SE25 (easy) \\
1910 \sudoku{se25.sud}
1911 \end{center}\end{minipage}
1912
1913 \renewcommand*{\puzzlefile}{se30.sud}
1914 \writepuzzle%
1915 {.3.12....}{1.87..4.6}{.98....1}%
1916 {5....7.3.}{.73.9.2.4}{.92..65..}%
1917 {.4...8.5.}{3...7...8}{2..4.596.}%
1918 [SSBL easy 30]
1919 \vfill
1920 \begin{minipage}{0.95\linewidth}\begin{center}
1921 SE30 (easy) \\
1922 \sudoku{se30.sud}
1923 \end{center}\end{minipage}

```

```

1924
1925 \renewcommand*{\puzzlefile}{tg5.sud}
1926 \writepuzzle%
1927 {.6.1.9.8.}{.19...74.}{2.....1}%
1928 {..79.26..}{.3.....2.}{..14.35..}%
1929 {1.....5}{.75...23.}{.9.5.6.7.}%
1930 [DT2 gentle 5]
1931 \vfill
1932 \begin{minipage}{0.95\linewidth}\begin{center}
1933 TG5 (gentle) \\
1934 \sudoku{tg5.sud}
1935 \end{center}\end{minipage}
1936
1937 \renewcommand*{\puzzlefile}{tg10.sud}
1938 \writepuzzle%
1939 {...96..4}{..1....2}{56...8...}%
1940 {2.8....9.}{9.63.52.7}{.3...4.6}%
1941 {...9...58}{7.....9..}{8..45....}%
1942 [DT2 gentle 10]
1943 \vfill
1944 \begin{minipage}{0.95\linewidth}\begin{center}
1945 TG10 (gentle) \\
1946 \sudoku{tg10.sud}
1947 \end{center}\end{minipage}
1948
1949 \renewcommand*{\puzzlefile}{tg15.sud}
1950 \writepuzzle%
1951 {...1.8...}{.57...18.}{98....26}%
1952 {..67419..}{.....}{..42536..}%
1953 {72.....94}{.19...73.}{...3.9...}%
1954 [DT2 gentle 15]
1955 \vfill
1956 \begin{minipage}{0.95\linewidth}\begin{center}
1957 TG15 (gentle) \\
1958 \sudoku{tg15.sud}
1959 \end{center}\end{minipage}
1960
1961 \renewcommand*{\puzzlefile}{tg20.sud}
1962 \writepuzzle%
1963 {...7.9...}{.8.3.6.5.}{7.9...6.8}%
1964 {..42.18..}{.6.....4.}{..35.41..}%
1965 {9.2...5.7}{.4.8.7.1.}{...1.2...}%
1966 [DT2 gentle 20]
1967 \vfill
1968 \begin{minipage}{0.95\linewidth}\begin{center}
1969 TG20 (gentle) \\
1970 \sudoku{tg20.sud}
1971 \end{center}\end{minipage}
1972
1973 \renewcommand*{\puzzlefile}{tg25.sud}

```

```

1974 \writepuzzle%
1975 {7..4..5..}{.61..5.7.}{.9.7.1..6}%
1976 {8...27...}{.....}{...34...1}%
1977 {1..8.2.3.}{.7.1..62.}{..3..9..7}%
1978 [DT2 gentle 25]
1979 \vfill
1980 \begin{minipage}{0.95\linewidth}\begin{center}
1981 TG25 (gentle) \\
1982 \sudoku{tg25.sud}
1983 \end{center}\end{minipage}
1984
1985 \renewcommand*\puzzlefile{tg30.sud}
1986 \writepuzzle%
1987 {37.4.....}{9....61.4}{...23...}%
1988 {.4.1..9..}{.12...79.}{.5..7.4.}%
1989 {...36....}{4.32...6}{.....9.57}%
1990 [DT2 gentle 30]
1991 \vfill
1992 \begin{minipage}{0.95\linewidth}\begin{center}
1993 TG30 (gentle) \\
1994 \sudoku{tg30.sud}
1995 \end{center}\end{minipage}
1996
1997 \renewcommand*\puzzlefile{st1226.sud}
1998 \writepuzzle%
1999 {9...4.31}{7452.....}{.8..674..}%
2000 {.26.4...5}{5..6.9..8}{4...2.37.}%
2001 {..759..6.}{.....1823}{21.3...7}%
2002 [Seattle Times 2005/12/26 1 star]
2003 \vfill
2004 \begin{minipage}{0.95\linewidth}\begin{center}
2005 ST1226 (1 star) \\
2006 \sudoku{st1226.sud}
2007 \end{center}\end{minipage}
2008
2009 \renewcommand*\puzzlefile{st0206.sud}
2010 \writepuzzle%
2011 {17.42...5}{..2.1..96}{849..3...}%
2012 {...718..}{.25...67.}{..826....}%
2013 {...5..749}{53..9.1..}{6...48.23}%
2014 [Seattle Times 2005/02/06 1 star]
2015 \vfill
2016 \begin{minipage}{0.95\linewidth}\begin{center}
2017 ST0206 (1 star) \\
2018 \sudoku{st0206.sud}
2019 \end{center}\end{minipage}
2020
2021 \renewcommand*\puzzlefile{st0213.sud}
2022 \writepuzzle%
2023 {.94.2...7}{.36...1.5}{28.35....}%

```

```

2024 {...92...5.}{6.21.49.8}{7...63..}%
2025 {...13.46}{1.5...78.}{8...4.29.}%
2026 [Seattle Times 2006/02/13 1 star]
2027 \vfill
2028 \begin{minipage}{0.95\linewidth}\begin{center}
2029 ST0213 (1 star) \\
2030 \sudoku{st0213.sud}
2031 \end{center}\end{minipage}
2032
2033 \renewcommand*{\puzzlefile}{cs1.sud}
2034 \writepuzzle%
2035 {.46..731.}{2.36.4...}{87.3.1.45}%
2036 {657...423}{.....}{.81...756}%
2037 {43.8.6..1}{7..2..5.4}{..25.386.}%
2038 [CreateSudoku 1 (level 2)]
2039 \vfill
2040 \begin{minipage}{0.95\linewidth}\begin{center}
2041 CS1 (level 2) \\
2042 \sudoku{cs1.sud}
2043 \end{center}\end{minipage}
2044
2045 \renewcommand*{\puzzlefile}{cs2.sud}
2046 \writepuzzle%
2047 {.7.3.265.}{...6..9.7}{64.9.8.2.}%
2048 {...9...765}{.....}{756...4.2}%
2049 {29.5...76}{...2.15.3}{.4..7.1.}%
2050 [CreateSudoku 2 (level 2)]
2051 \vfill
2052 \begin{minipage}{0.95\linewidth}\begin{center}
2053 CS2 (level 2) \\
2054 \sudoku{cs2.sud}
2055 \end{center}\end{minipage}
2056
2057 \renewcommand*{\puzzlefile}{cs3.sud}
2058 \writepuzzle%
2059 {.5.8.71..}{..35.....}{78.2.1.56}%
2060 {.34...8.1}{.....}{8.1...56.}%
2061 {.4.7.6.12}{...4.36.8}{.781..34.}%
2062 [CreateSudoku 3 (level 2)]
2063 \vfill
2064 \begin{minipage}{0.95\linewidth}\begin{center}
2065 CS3 (level 2) \\
2066 \sudoku{cs3.sud}
2067 \end{center}\end{minipage}
2068
2069 %%%%%%%%%%%%%%% moderate
2070
2071 \clearpage
2072 \markboth{MODERATE}{MODERATE}
2073

```

```

2074 \renewcommand*{\puzzlefile}{sm5.sud}
2075 \writepuzzle%
2076 {..76..24.}{.46....9.}{92..85...}%
2077 {8....29..}{..3...4..}{..49....6}%
2078 {...73..24}{.5....86.}{.32..67..}%
2079 [SSBL medium 5]
2080 \vfill
2081 \begin{minipage}{0.95\linewidth}\begin{center}
2082 SM5 (medium) \\
2083 \sudoku{sm5.sud}
2084 \end{center}\end{minipage}
2085
2086 \renewcommand*{\puzzlefile}{sm10.sud}
2087 \writepuzzle%
2088 {..57.82..}{4.....6}{.7..6..8.}%
2089 {52.9.4.31}{.....}{86.1.7.54}%
2090 {.9..3..6.}{7.....5}{..24.61..}%
2091 [SSBL medium 10]
2092 \vfill
2093 \begin{minipage}{0.95\linewidth}\begin{center}
2094 SM10 (medium) \\
2095 \sudoku{sm10.sud}
2096 \end{center}\end{minipage}
2097
2098 \renewcommand*{\puzzlefile}{sm15.sud}
2099 \writepuzzle%
2100 {85...4..2}{4...2.9..}{...8...7.}%
2101 {...6....1}{.4..8..3.}{2....5..}%
2102 {.34..7...}{.87.9...6}{6..1...47}%
2103 [SSBL medium 15]
2104 \vfill
2105 \begin{minipage}{0.95\linewidth}\begin{center}
2106 SM15 (medium) \\
2107 \sudoku{sm15.sud}
2108 \end{center}\end{minipage}
2109
2110 \renewcommand*{\puzzlefile}{sm20.sud}
2111 \writepuzzle%
2112 {...34.2.97}{..4....2.}{.7.6....8}%
2113 {...9....}{.38...47.}{...1....}%
2114 {4...5.1.}{.5....6..}{72.3.19..}%
2115 [SSBL medium 20]
2116 \vfill
2117 \begin{minipage}{0.95\linewidth}\begin{center}
2118 SM20 (medium) \\
2119 \sudoku{sm20.sud}
2120 \end{center}\end{minipage}
2121
2122 \renewcommand*{\puzzlefile}{sm25.sud}
2123 \writepuzzle%

```

```

2124 {...4.79...}{7.6...2...}{.5..83.6}%
2125 {56..298...}{.8..1...}{.3.7....5}%
2126 {.....38}{....93...}{.4.5.....}%
2127 [SSBL medium 25]
2128 \vfill
2129 \begin{minipage}{0.95\linewidth}\begin{center}
2130 SM25 (medium) \\
2131 \sudoku{sm25.sud}
2132 \end{center}\end{minipage}
2133
2134 \renewcommand*{\puzzlefile}{sm30.sud}
2135 \writepuzzle%
2136 {...58.....}{...79...}{.461..7.}%
2137 {.5.3....4}{.23...65.}{9....8.3.}%
2138 {.3..572..}{...42....}{.....61..}%
2139 [SSBL medium 30]
2140 \vfill
2141 \begin{minipage}{0.95\linewidth}\begin{center}
2142 SM30 (medium) \\
2143 \sudoku{sm30.sud}
2144 \end{center}\end{minipage}
2145
2146 \renewcommand*{\puzzlefile}{tm35.sud}
2147 \writepuzzle%
2148 {5..3.8..6}{.4.5.8..}{2.8...1.5}%
2149 {...62.74..}{.....}{.94.32..}%
2150 {9.5...6.4}{.2.1.9..}{1..8.9..2}%
2151 [DT2 moderate 35]
2152 \vfill
2153 \begin{minipage}{0.95\linewidth}\begin{center}
2154 TM35 (moderate) \\
2155 \sudoku{tm35.sud}
2156 \end{center}\end{minipage}
2157
2158 \renewcommand*{\puzzlefile}{tm40.sud}
2159 \writepuzzle%
2160 {...1...92}{9...7..36}{3...2...}%
2161 {.3...7.9}{.8.9.5.6.}{4.5...8.}%
2162 {...8...5}{54..2...7}{12...3...}%
2163 [DT2 moderate 40]
2164 \vfill
2165 \begin{minipage}{0.95\linewidth}\begin{center}
2166 TM40 (moderate) \\
2167 \sudoku{tm40.sud}
2168 \end{center}\end{minipage}
2169
2170 \renewcommand*{\puzzlefile}{tm45.sud}
2171 \writepuzzle%
2172 {.1.2.8.3.}{.87.94..}{9.....6}%
2173 {...9.8.5..}{6.....3}{.3.5.6..}%

```

```

2174 {2.....7}{..73.51..}{.6.4.7.9.}%
2175 [DT2 moderate 45]
2176 \vfill
2177 \begin{minipage}{0.95\linewidth}\begin{center}
2178 TM45 (moderate) \\
2179 \sudoku{tm45.sud}
2180 \end{center}\end{minipage}
2181
2182 \renewcommand*\puzzlefile{tm50.sud}
2183 \writepuzzle%
2184 {...9.2..4}{4..3.5...}{3....7.6}%
2185 {6..8..3..}{1.3...9.7}{..9..7..2}%
2186 {9.5.....8}{...2.9..1}{2..7.4...}%
2187 [DT2 moderate 50]
2188 \vfill
2189 \begin{minipage}{0.95\linewidth}\begin{center}
2190 TM50 (moderate) \\
2191 \sudoku{tm50.sud}
2192 \end{center}\end{minipage}
2193
2194 \renewcommand*\puzzlefile{tm55.sud}
2195 \writepuzzle%
2196 {7.....}{24.1.8...}{.6..72..3}%
2197 {...4.29.8.}{.7.....6.}{.9.46.7..}%
2198 {1..68..3.}{...3.7.91}{.....6}%
2199 [DT2 moderate 55]
2200 \vfill
2201 \begin{minipage}{0.95\linewidth}\begin{center}
2202 TM55 (moderate) \\
2203 \sudoku{tm55.sud}
2204 \end{center}\end{minipage}
2205
2206 \renewcommand*\puzzlefile{tm60.sud}
2207 \writepuzzle%
2208 {724.9.5..}{.5..2...8}{.8.5.7...}%
2209 {...7..64.}{...8...}{.79..6...}%
2210 {...8.5.2.}{6...4..8.}{..8.3.174}%
2211 [DT2 moderate 60]
2212 \vfill
2213 \begin{minipage}{0.95\linewidth}\begin{center}
2214 TM60 (moderate) \\
2215 \sudoku{tm60.sud}
2216 \end{center}\end{minipage}
2217
2218 \renewcommand*\puzzlefile{st0914.sud}
2219 \writepuzzle%
2220 {.6.29..8.}{..8....3.}{.1..78...}%
2221 {.217.9..8}{6.....3}{7..6.492.}%
2222 {...12..4.}{.7...3..}{.5..86.7.}%
2223 [Seattle Times 2005/09/14 2 stars]

```

```

2224 \vfill
2225 \begin{minipage}{0.95\linewidth}\begin{center}
2226 ST0914 (2 stars) \\
2227 \sudoku{st0914.sud}
2228 \end{center}\end{minipage}
2229
2230 \renewcommand*{\puzzlefile}{st0207.sud}
2231 \writepuzzle%
2232 {...46}{5..8.9.73}{...6..5.1}%
2233 {.592..1..}{6...1...7}{..3..785.}%
2234 {2.5..6...}{19.7.5..8}{34.....}%
2235 [Seattle Times 2005/02/07 2 stars]
2236 \vfill
2237 \begin{minipage}{0.95\linewidth}\begin{center}
2238 ST0207 (2 stars) \\
2239 \sudoku{st0207.sud}
2240 \end{center}\end{minipage}
2241
2242 \renewcommand*{\puzzlefile}{cs4.sud}
2243 \writepuzzle%
2244 {...4.597.}{..67.83.2}{78.1.2.45}%
2245 {891...75.}{.....}{567...423}%
2246 {.1...4.67}{..89.15..}{.4...29.}%
2247 [CreateSudoku 4 (level 8)]
2248 \vfill
2249 \begin{minipage}{0.95\linewidth}\begin{center}
2250 CS4 (level 4) \\
2251 \sudoku{cs4.sud}
2252 \end{center}\end{minipage}
2253
2254 \renewcommand*{\puzzlefile}{cs5.sud}
2255 \writepuzzle%
2256 {...6..978}{..498...2}{.97.21...}%
2257 {...84.27..}{.7.....23}{3...6...9}%
2258 {7.6..95..}{...87...1}{.2..4.867}%
2259 [CreateSudoku 5 (level 4)]
2260 \vfill
2261 \begin{minipage}{0.95\linewidth}\begin{center}
2262 CS5 (level 4) \\
2263 \sudoku{cs5.sud}
2264 \end{center}\end{minipage}
2265
2266 \renewcommand*{\puzzlefile}{cs6.sud}
2267 \writepuzzle%
2268 {.126.4.7.}{6.5..73.2}{.7.3.1.4.}%
2269 {756...423}{.....}{18.....6}%
2270 {.3.8.6..1}{2.1.....7}{..72...3.}%
2271 [CreateSudoku 6 (level 4)]
2272 \vfill
2273 \begin{minipage}{0.95\linewidth}\begin{center}

```

```

2274 CS6 (level 4) \\
2275 \sudoku{cs6.sud}
2276 \end{center}\end{minipage}
2277
2278 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% hard
2279
2280 \clearpage
2281 \markboth{HARD}{HARD}
2282
2283 \renewcommand*\puzzlefile{sh5.sud}
2284 \writepuzzle%
2285 {.9..5...}{47...2...}{.836.4.1.}%
2286 {.15....8.}{...3.7...}{.2....74.}%
2287 {4.5.187.}{...7...52}{...2..1..}%
2288 [SSBL hard 5]
2289 \vfill
2290 \begin{minipage}{0.95\linewidth}\begin{center}
2291 SH5 (hard) \\
2292 \sudoku{sh5.sud}
2293 \end{center}\end{minipage}
2294
2295 \renewcommand*\puzzlefile{sh10.sud}
2296 \writepuzzle%
2297 {3.921..5}{15.83....}{.....}%
2298 {59.....4}{37.....81}{6.....53}%
2299 {.....}{...19.26}{9..574.1.}%
2300 [SSBL hard 10]
2301 \vfill
2302 \begin{minipage}{0.95\linewidth}\begin{center}
2303 SH10 (hard) \\
2304 \sudoku{sh10.sud}
2305 \end{center}\end{minipage}
2306
2307 \renewcommand*\puzzlefile{sh15.sud}
2308 \writepuzzle%
2309 {2..9.....}{7..6..59}{...4...61}%
2310 {7.1..2..4}{...4....}{8..1..9.6}%
2311 {12...3...}{53..2..7.}{.....9..2}%
2312 [SSBL hard 15]
2313 \vfill
2314 \begin{minipage}{0.95\linewidth}\begin{center}
2315 SH15 (hard) \\
2316 \sudoku{sh15.sud}
2317 \end{center}\end{minipage}
2318
2319 \renewcommand*\puzzlefile{sh20.sud}
2320 \writepuzzle%
2321 {6.2...3.}{9...3...1}{...9.2..}%
2322 {1..7.....}{32...46.}{.....4..2}%
2323 {..7.4....}{2...5...8}{4...6.5}%

```

```

2324 [SSBL hard 20]
2325 \vfill
2326 \begin{minipage}{0.95\linewidth}\begin{center}
2327 SH20 (hard) \\
2328 \sudoku{sh20.sud}
2329 \end{center}\end{minipage}
2330
2331 \renewcommand*\puzzlefile{sh25.sud}
2332 \writepuzzle%
2333 {4...1..8}{..7.3845.}{..24...1.}%
2334 {28.9.....}{.4...5..}{....4.86}%
2335 {.4...92..}{.2134.8..}{9..6....1}%
2336 [SSBL hard 25]
2337 \vfill
2338 \begin{minipage}{0.95\linewidth}\begin{center}
2339 SH25 (hard) \\
2340 \sudoku{sh25.sud}
2341 \end{center}\end{minipage}
2342
2343 \renewcommand*\puzzlefile{sh30.sud}
2344 \writepuzzle%
2345 {...5..9.}{.53..78..}{78.4.....}%
2346 {3...9.2.}{.72...38.}{.1.3....4}%
2347 {...1.35}{..58..41.}{.2..3....}%
2348 [SSBL hard 30]
2349 \vfill
2350 \begin{minipage}{0.95\linewidth}\begin{center}
2351 SH30 (hard) \\
2352 \sudoku{sh30.sud}
2353 \end{center}\end{minipage}
2354
2355 \renewcommand*\puzzlefile{tt100.sud}
2356 \writepuzzle%
2357 {...896..7}{..23.5...}{.8...5.1}%
2358 {15...9...}{3.....9}{...6...54}%
2359 {8.1...7..}{...9.18..}{6..428...}%
2360 [DT2 tough 100]
2361 \vfill
2362 \begin{minipage}{0.95\linewidth}\begin{center}
2363 TT100 (tough) \\
2364 \sudoku{tt100.sud}
2365 \end{center}\end{minipage}
2366
2367 \renewcommand*\puzzlefile{tt105.sud}
2368 \writepuzzle%
2369 {...5.7..46}{6..25....}{.3...6...}%
2370 {2..7.....}{.78...35.}{.....1..7}%
2371 {...8...2.}{...97..5}{81..2.9..}%
2372 [DT2 tough 105]
2373 \vfill

```

```

2374 \begin{minipage}{0.95\linewidth}\begin{center}
2375 TT105 (tough) \\
2376 \sudoku{tt105.sud}
2377 \end{center}\end{minipage}
2378
2379 \renewcommand*{\puzzlefile}{tt110.sud}
2380 \writepuzzle%
2381 {.79.1....}{.4.5....7}{1..6.....}%
2382 {9368..1..}{.4...9..}{..7..4632}%
2383 {....8..1}{5....6.9.}{....4.82.}%
2384 [DT2 tough 110]
2385 \vfill
2386 \begin{minipage}{0.95\linewidth}\begin{center}
2387 TT110 (tough) \\
2388 \sudoku{tt110.sud}
2389 \end{center}\end{minipage}
2390
2391 \renewcommand*{\puzzlefile}{tt115.sud}
2392 \writepuzzle%
2393 {432.5....}{....3...}{75.1.....}%
2394 {.5.823..}{.6.5.1.4.}{..193.2..}%
2395 {....6.58}{...3.....}{....2.934}%
2396 [DT2 tough 115]
2397 \vfill
2398 \begin{minipage}{0.95\linewidth}\begin{center}
2399 TT115 (tough) \\
2400 \sudoku{tt115.sud}
2401 \end{center}\end{minipage}
2402
2403 \renewcommand*{\puzzlefile}{tt120.sud}
2404 \writepuzzle%
2405 {.8.1.7..}{.4.67.8..}{..2..461.}%
2406 {.9...1.8}{.....}{2.1...9.}%
2407 {.759..4.}{.4.52.6.}{.9.3.5..}%
2408 [DT2 tough 120]
2409 \vfill
2410 \begin{minipage}{0.95\linewidth}\begin{center}
2411 TT120 (tough) \\
2412 \sudoku{tt120.sud}
2413 \end{center}\end{minipage}
2414
2415 \renewcommand*{\puzzlefile}{st1130.sud}
2416 \writepuzzle%
2417 {8..356...}{5....4.1.}{34.....}%
2418 {7.156..2.}{6.....7.}{2..796.5}%
2419 {.....32}{.9.4...8}{...682..9}%
2420 [Seattle Times 2005/11/30 3 stars]
2421 \vfill
2422 \begin{minipage}{0.95\linewidth}\begin{center}
2423 ST1130 (3 stars) \\

```

```

2424 \sudoku{st1130.sud}
2425 \end{center}\end{minipage}
2426
2427 \renewcommand*{\puzzlefile}{st1207.sud}
2428 \writepuzzle%
2429 {...483..6}{.....57.2}{.....38.}%
2430 {...8..16..}{.26.5.17.}{.57..2..}%
2431 {.82.....}{9.18.....}{4..527...}%
2432 [Seattle Times 2005/12/07 3 stars]
2433 \vfill
2434 \begin{minipage}{0.95\linewidth}\begin{center}
2435 ST1207 (3 stars) \\
2436 \sudoku{st1207.sud}
2437 \end{center}\end{minipage}
2438
2439 \renewcommand*{\puzzlefile}{st1209.sud}
2440 \writepuzzle%
2441 {...6.....}{.4..8..5.}{.519..2.8}%
2442 {...729...3}{.9.....1.}{4...679..}%
2443 {6.8..957.}{.7..2..9.}{.....4...}%
2444 [Seattle Times 2005/12/09 5 stars]
2445 \vfill
2446 \begin{minipage}{0.95\linewidth}\begin{center}
2447 ST1209 (5 stars) \\
2448 \sudoku{st1209.sud}
2449 \end{center}\end{minipage}
2450
2451 \renewcommand*{\puzzlefile}{st1221.sud}
2452 \writepuzzle%
2453 {.....392.}{4....5.7.}{29.4...63}%
2454 {...4....5.}{...2.1...}{.5....3..}%
2455 {74...2.96}{.6.5....7}{.217.....}%
2456 [Seattle Times 2005/12/21 3 stars]
2457 \vfill
2458 \begin{minipage}{0.95\linewidth}\begin{center}
2459 ST1221 (3 stars) \\
2460 \sudoku{st1221.sud}
2461 \end{center}\end{minipage}
2462
2463 \renewcommand*{\puzzlefile}{st1228.sud}
2464 \writepuzzle%
2465 {6..1..5..}{.....379}{.942.....}%
2466 {...5...8.3}{.4.....9.}{9.7...6..}%
2467 {.....623.}{138.....}{.6..8..7}%
2468 [Seattle Times 2005/12/28 3 stars]
2469 \vfill
2470 \begin{minipage}{0.95\linewidth}\begin{center}
2471 ST1228 (3 stars) \\
2472 \sudoku{st1228.sud}
2473 \end{center}\end{minipage}

```

```

2474
2475 \renewcommand*{\puzzlefile}{st0126.sud}
2476 \writepuzzle%
2477 {.36...7.}{1...352..}{.4.....}%
2478 {.6....29}{7.....4}{32...5..}%
2479 {.....1..}{.824...5}{.4...93..}%
2480 [Seattle Times 2006/01/26 4 stars]
2481 \vfill
2482 \begin{minipage}{0.95\linewidth}\begin{center}
2483 ST0126 (4 stars) \\
2484 \sudoku{st0126.sud}
2485 \end{center}\end{minipage}
2486
2487 \renewcommand*{\puzzlefile}{st0127.sud}
2488 \writepuzzle%
2489 {25.8.....}{...9.8.2}{9..4.7...}%
2490 {64.....9.}{.9.1.3.4.}{.7....21}%
2491 {...5.4..7}{7.6.8....}{.....1.83}%
2492 [Seattle Times 2006/01/27 5 stars]
2493 \vfill
2494 \begin{minipage}{0.95\linewidth}\begin{center}
2495 ST0127 (5 stars) \\
2496 \sudoku{st0127.sud}
2497 \end{center}\end{minipage}
2498
2499 \renewcommand*{\puzzlefile}{st0204.sud}
2500 \writepuzzle%
2501 {.5..2.8.6}{.....6..3}{.6..132..}%
2502 {.7...96.}{4.....1}{.16...3.}%
2503 {.723..8.}{6..7.....}{2.4.9..7.}%
2504 [Seattle Times 2006/02/04 6 stars]
2505 \vfill
2506 \begin{minipage}{0.95\linewidth}\begin{center}
2507 ST0204 (6 stars) \\
2508 \sudoku{st0204.sud}
2509 \end{center}\end{minipage}
2510
2511 \renewcommand*{\puzzlefile}{st0208.sud}
2512 \writepuzzle%
2513 {21.9...7}{5.9..421.}{.7.....}%
2514 {.875....}{3.14.95.8}{...361..}%
2515 {.....4..}{.325..9.1}{1...8.25}%
2516 [Seattle Times 2005/02/08 3 stars]
2517 \vfill
2518 \begin{minipage}{0.95\linewidth}\begin{center}
2519 ST0208 (3 stars) \\
2520 \sudoku{st0208.sud}
2521 \end{center}\end{minipage}
2522
2523 \renewcommand*{\puzzlefile}{st0209.sud}

```

```

2524 \writepuzzle%
2525 {..9.....}{.523.1...}{8.12.6...}%
2526 {..4....1.}{97..2..84}{.2....9..}%
2527 {...8.41.5}{...6.524.}{.....8..}%
2528 [Seattle Times 2005/02/09 4 stars]
2529 \vfill
2530 \begin{minipage}{0.95\linewidth}\begin{center}
2531 ST0209 (4 stars) \\
2532 \sudoku{st0209.sud}
2533 \end{center}\end{minipage}
2534
2535 %%%%%%%%%%% answers
2536
2537 \clearpage
2538 \pagestyle{empty}
2539 \cleardoublepage
2540 \pagestyle{myheadings}
2541 \markboth{SOLUTIONS}{SOLUTIONS}
2542
2543 \cluefont{\normalsize}
2544 \cellsize{1.5\baselineskip}
2545
2546 \renewcommand*{\puzzlefile}{se5.ans}
2547 \writepuzzle%
2548 {724195863}{561283497}{938647251}%
2549 {189756324}{256314789}{473928615}%
2550 {645839172}{892471536}{317562948}%
2551 [SSBL easy 5 solution]
2552 \renewcommand*{\puzzlefile}{se10.ans}
2553 \writepuzzle%
2554 {156349782}{482715639}{793286145}%
2555 {649853271}{217964358}{835172496}%
2556 {524697813}{371528964}{968431527}%
2557 [SSBL easy 10 solution]
2558 \vfill
2559 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2560 SE5 (easy solution) \\
2561 \sudoku{se5.ans}
2562 \end{center}\end{minipage}
2563 \hfill
2564 \begin{minipage}{0.47\linewidth}\begin{center}
2565 SE10 (easy solution) \\
2566 \sudoku{se10.ans}
2567 \end{center}\end{minipage}
2568
2569 \renewcommand*{\puzzlefile}{se15.ans}
2570 \writepuzzle%
2571 {456713982}{231698475}{879254631}%
2572 {597846213}{124935867}{368172549}%
2573 {643527198}{712489356}{985361724}%

```

```

2574 [SSBL easy 15 solution]
2575 \renewcommand*{\puzzlefile}{se20.ans}
2576 \writepuzzle%
2577 {368912547}{975438621}{412657839}%
2578 {291543768}{783296414}{546781293}%
2579 {657129384}{134865972}{829374156}%
2580 [SSBL easy 20 solution]
2581 \vfill
2582 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2583 SE15 (easy solution) \\
2584 \sudoku{se15.ans}
2585 \end{center}\end{minipage}
2586 \hfill
2587 \begin{minipage}{0.47\linewidth}\begin{center}
2588 SE20 (easy solution) \\
2589 \sudoku{se20.ans}
2590 \end{center}\end{minipage}
2591
2592 \renewcommand*{\puzzlefile}{se25.ans}
2593 \writepuzzle%
2594 {278963154}{645712839}{931854276}%
2595 {823641795}{159387462}{764295381}%
2596 {482579613}{397126548}{516438927}%
2597 [SSBL easy 25 solution]
2598 \renewcommand*{\puzzlefile}{se30.ans}
2599 \writepuzzle%
2600 {436129875}{128753496}{759864321}%
2601 {514287639}{673591284}{892346517}%
2602 {941638752}{365972148}{287415963}%
2603 [SSBL easy 30 solution]
2604 \vfill
2605 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2606 SE25 (easy solution) \\
2607 \sudoku{se25.ans}
2608 \end{center}\end{minipage}
2609 \hfill
2610 \begin{minipage}{0.47\linewidth}\begin{center}
2611 SE30 (easy solution) \\
2612 \sudoku{se30.ans}
2613 \end{center}\end{minipage}
2614
2615 \renewcommand*{\puzzlefile}{tg5.ans}
2616 \writepuzzle%
2617 {764159382}{519238746}{823674951}%
2618 {457982613}{936715428}{281463597}%
2619 {142397865}{675841239}{398526174}%
2620 [DT2 gentle 5 solution]
2621 \renewcommand*{\puzzlefile}{tg10.ans}
2622 \writepuzzle%
2623 {382196574}{491573862}{567248139}%

```

```

2624 {278614395}{946385217}{135729486}%
2625 {624931758}{753862941}{819457623}%
2626 [DT2 gentle 10 solution]
2627 \vfill
2628 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2629 TG5 (gentle solution) \\
2630 \sudoku{tg5.ans}
2631 \end{center}\end{minipage}
2632 \hfill
2633 \begin{minipage}{0.47\linewidth}\begin{center}
2634 TG10 (gentle solution) \\
2635 \sudoku{tg10.ans}
2636 \end{center}\end{minipage}
2637
2638 \renewcommand*{\puzzlefile}{tg15.ans}
2639 \writepuzzle%
2640 {462198573}{357462189}{981537426}%
2641 {236741958}{175986342}{894253617}%
2642 {723615894}{619824735}{548379261}%
2643 [DT2 gentle 15 solution]
2644 \renewcommand*{\puzzlefile}{tg20.ans}
2645 \writepuzzle%
2646 {256789431}{481326759}{739415628}%
2647 {594261873}{167938245}{823574196}%
2648 {912643587}{645897312}{378152964}%
2649 [DT2 gentle 20 solution]
2650 \vfill
2651 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2652 TG15 (gentle solution) \\
2653 \sudoku{tg15.ans}
2654 \end{center}\end{minipage}
2655 \hfill
2656 \begin{minipage}{0.47\linewidth}\begin{center}
2657 TG20 (gentle solution) \\
2658 \sudoku{tg20.ans}
2659 \end{center}\end{minipage}
2660
2661 \renewcommand*{\puzzlefile}{tg25.ans}
2662 \writepuzzle%
2663 {782463519}{461295378}{395781246}%
2664 {819527463}{634918752}{527346891}%
2665 {256872934}{978134625}{243659187}%
2666 [DT2 gentle 25 solution]
2667 \renewcommand*{\puzzlefile}{tg30.ans}
2668 \writepuzzle%
2669 {376418529}{928756134}{154923678}%
2670 {749132865}{612584793}{835697241}%
2671 {597361482}{483275916}{261849357}%
2672 [DT2 gentle 30 solution]
2673 \vfill

```

```

2674 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2675 TG25 (gentle solution) \\
2676 \sudoku{tg25.ans}
2677 \end{center}\end{minipage}
2678 \hfill
2679 \begin{minipage}{0.47\linewidth}\begin{center}
2680 TG30 (gentle solution) \\
2681 \sudoku{tg30.ans}
2682 \end{center}\end{minipage}
2683
2684 \renewcommand*{\puzzlefile}{st1226.ans}
2685 \writepuzzle%
2686 {962854731}{745213689}{183967452}%
2687 {326748915}{571639248}{498125376}%
2688 {837592164}{659471823}{214386597}%
2689 [Seattle Times 2005/12/26 1 star solution]
2690 \renewcommand*{\puzzlefile}{st0206.ans}
2691 \writepuzzle%
2692 {176429385}{352817496}{849653217}%
2693 {463971852}{925384671}{718265934}%
2694 {281536749}{534792168}{697148523}%
2695 [Seattle Times 2005/02/06 1 star solution]
2696 \vfill
2697 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2698 ST1226 (1 star solution) \\
2699 \sudoku{st1226.ans}
2700 \end{center}\end{minipage}
2701 \hfill
2702 \begin{minipage}{0.47\linewidth}\begin{center}
2703 ST0206 (1 star solution) \\
2704 \sudoku{st0206.ans}
2705 \end{center}\end{minipage}
2706
2707 \renewcommand*{\puzzlefile}{st0123.ans}
2708 \writepuzzle%
2709 {594621837}{736489125}{281357469}%
2710 {319278654}{652134978}{478596312}%
2711 {927813546}{145962783}{863745291}%
2712 [Seattle Times 2006/01/23 1 star solution]
2713 \renewcommand*{\puzzlefile}{cs1.ans}
2714 \writepuzzle%
2715 {546987312}{213654978}{879321645}%
2716 {657198423}{324765189}{981432756}%
2717 {435876291}{768219534}{192543867}%
2718 [CreateSudoku 1 (level 2) solution]
2719 \vfill
2720 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2721 ST0123 (1 star solution) \\
2722 \sudoku{st0123.ans}
2723 \end{center}\end{minipage}

```

```

2724 \hfill
2725 \begin{minipage}{0.47\linewidth}\begin{center}
2726 CS1 (level 2 solution) \\
2727 \sudoku{cs1.ans}
2728 \end{center}\end{minipage}
2729
2730 \renewcommand*{\puzzlefile}{cs2.ans}
2731 \writepuzzle%
2732 {978312654}{312645987}{645978321}%
2733 {189423765}{423756198}{756189432}%
2734 {291534876}{867291543}{534867219}%
2735 [CreateSudoku 2 (level 2) solution]
2736 \renewcommand*{\puzzlefile}{cs3.ans}
2737 \writepuzzle%
2738 {456897123}{123564789}{789231456}%
2739 {234675891}{567918234}{891342567}%
2740 {345786912}{912453678}{678129345}%
2741 [CreateSudoku 3 (level 2) solution]
2742 \vfill
2743 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2744 CS2 (level 2 solution) \\
2745 \sudoku{cs2.ans}
2746 \end{center}\end{minipage}
2747 \hfill
2748 \begin{minipage}{0.47\linewidth}\begin{center}
2749 CS3 (level 2 solution) \\
2750 \sudoku{cs3.ans}
2751 \end{center}\end{minipage}
2752
2753 %%%%%%%%%%%%%%% medium
2754
2755 \renewcommand*{\puzzlefile}{sm5.ans}
2756 \writepuzzle%
2757 {387691245}{546273198}{921485637}%
2758 {865342971}{193567482}{274918356}%
2759 {618739524}{759124863}{432856719}%
2760 [SSBL medium 5 solution]
2761 \renewcommand*{\puzzlefile}{sm10.ans}
2762 \writepuzzle%
2763 {635748219}{418392576}{279561483}%
2764 {527984631}{941653827}{863127954}%
2765 {194835762}{786219345}{352476198}%
2766 [SSBL medium 10 solution]
2767 \vfill
2768 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2769 SM5 (medium solution) \\
2770 \sudoku{sm5.ans}
2771 \end{center}\end{minipage}
2772 \hfill
2773 \begin{minipage}{0.47\linewidth}\begin{center}

```

```

2774 SM10 (medium solution) \\
2775 \sudoku{sm10.ans}
2776 \end{center}\end{minipage}
2777
2778 \renewcommand*{\puzzlefile}{sm15.ans}
2779 \writepuzzle%
2780 {853974612}{471326958}{962815473}%
2781 {396542781}{745681239}{218739564}%
2782 {134267895}{587493126}{629158347}%
2783 [SSBL medium 15 solution]
2784 \renewcommand*{\puzzlefile}{sm20.ans}
2785 \writepuzzle%
2786 {863452197}{594178326}{172639548}%
2787 {617894235}{938526471}{234713869}%
2788 {489265713}{351947682}{726381954}%
2789 [SSBL medium 20 solution]
2790 \vfill
2791 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2792 SM15 (medium solution) \\
2793 \sudoku{sm15.ans}
2794 \end{center}\end{minipage}
2795 \hfill
2796 \begin{minipage}{0.47\linewidth}\begin{center}
2797 SM20 (medium solution) \\
2798 \sudoku{sm20.ans}
2799 \end{center}\end{minipage}
2800
2801 \renewcommand*{\puzzlefile}{sm25.ans}
2802 \writepuzzle%
2803 {382467951}{716935284}{495218376}%
2804 {564329817}{278651493}{139784625}%
2805 {927146538}{651893742}{843572169}%
2806 [SSBL medium 25 solution]
2807 \renewcommand*{\puzzlefile}{sm30.ans}
2808 \writepuzzle%
2809 {765834921}{318279465}{294615378}%
2810 {657392814}{823741659}{941568732}%
2811 {439157286}{186423597}{572986143}%
2812 [SSBL medium 30 solution]
2813 \vfill
2814 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2815 SM25 (medium solution) \\
2816 \sudoku{sm25.ans}
2817 \end{center}\end{minipage}
2818 \hfill
2819 \begin{minipage}{0.47\linewidth}\begin{center}
2820 SM30 (medium solution) \\
2821 \sudoku{sm30.ans}
2822 \end{center}\end{minipage}
2823

```

```

2824 \renewcommand*{\puzzlefile}{tm35.ans}
2825 \writepuzzle%
2826 {591328746}{674951823}{238674195}%
2827 {356297481}{427185369}{819463257}%
2828 {985732614}{742516938}{163849572}%
2829 [DT2 moderate 35 solution]
2830 \renewcommand*{\puzzlefile}{tm40.ans}
2831 \writepuzzle%
2832 {874136592}{952478136}{316592874}%
2833 {631284759}{287915463}{495367281}%
2834 {763841925}{548629317}{129753648}%
2835 [DT2 moderate 40 solution]
2836 \vfill
2837 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2838 TM35 (moderate solution) \\
2839 \sudoku{tm35.ans}
2840 \end{center}\end{minipage}
2841 \hfill
2842 \begin{minipage}{0.47\linewidth}\begin{center}
2843 TM40 (moderate solution) \\
2844 \sudoku{tm40.ans}
2845 \end{center}\end{minipage}
2846
2847 \renewcommand*{\puzzlefile}{tm45.ans}
2848 \writepuzzle%
2849 {716248935}{538769421}{924531786}%
2850 {179683542}{652974813}{483152679}%
2851 {241896357}{897325164}{365417298}%
2852 [DT2 moderate 45 solution]
2853 \renewcommand*{\puzzlefile}{tm50.ans}
2854 \writepuzzle%
2855 {751962834}{468375219}{392148756}%
2856 {627891345}{143526987}{589437162}%
2857 {975613428}{834259671}{216784593}%
2858 [DT2 moderate 50 solution]
2859 \vfill
2860 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2861 TM45 (moderate solution) \\
2862 \sudoku{tm45.ans}
2863 \end{center}\end{minipage}
2864 \hfill
2865 \begin{minipage}{0.47\linewidth}\begin{center}
2866 TM50 (moderate solution) \\
2867 \sudoku{tm50.ans}
2868 \end{center}\end{minipage}
2869
2870 \renewcommand*{\puzzlefile}{tm55.ans}
2871 \writepuzzle%
2872 {731946528}{245138679}{968572143}%
2873 {614729385}{572813964}{893465712}%

```

```

2874 {159684237}{426357891}{387291456}%
2875 [DT2 moderate 55 solution]
2876 \renewcommand*{\puzzlefile}{tm60.ans}
2877 \writepuzzle%
2878 {734198562}{156324798}{982567431}%
2879 {813752649}{465983217}{279416853}%
2880 {341875926}{697241385}{528639174}%
2881 [DT2 moderate 60 solution]
2882 \vfill
2883 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2884 TM55 (moderate solution) \\
2885 \sudoku{tm55.ans}
2886 \end{center}\end{minipage}
2887 \hfill
2888 \begin{minipage}{0.47\linewidth}\begin{center}
2889 TM60 (moderate solution) \\
2890 \sudoku{tm60.ans}
2891 \end{center}\end{minipage}
2892
2893 \renewcommand*{\puzzlefile}{st0914.ans}
2894 \writepuzzle%
2895 {567293184}{298461537}{413578692}%
2896 {321759468}{649812753}{785634921}%
2897 {936127845}{872945316}{154386279}%
2898 [Seattle Times 2005/09/14 2 star solution]
2899 \renewcommand*{\puzzlefile}{st0207.ans}
2900 \writepuzzle%
2901 {872351946}{561849273}{934672581}%
2902 {759283164}{628514397}{413967852}%
2903 {285436719}{196725438}{347198625}%
2904 [Seattle Times 2005/02/07 2 star solution]
2905 \vfill
2906 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2907 ST0914 (2 star solution) \\
2908 \sudoku{st0914.ans}
2909 \end{center}\end{minipage}
2910 \hfill
2911 \begin{minipage}{0.47\linewidth}\begin{center}
2912 ST0207 (2 star solution) \\
2913 \sudoku{st0207.ans}
2914 \end{center}\end{minipage}
2915
2916 \renewcommand*{\puzzlefile}{cs4.ans}
2917 \writepuzzle%
2918 {123465978}{456798312}{789132645}%
2919 {891243756}{234576189}{567819423}%
2920 {912354867}{678921534}{345687291}%
2921 [CreateSudoku 4 (level 4) solution]
2922 \renewcommand*{\puzzlefile}{cs5.ans}
2923 \writepuzzle%

```

```

2924 {231654978}{564987312}{897321645}%
2925 {918432756}{675198423}{342765189}%
2926 {786219534}{453876291}{129543867}%
2927 [CreateSudoku 5 (level 4) solution]
2928 \vfill
2929 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2930 CS4 (level 8 solution) \\
2931 \sudoku{cs4.ans}
2932 \end{center}\end{minipage}
2933 \hfill
2934 \begin{minipage}{0.47\linewidth}\begin{center}
2935 CS5 (level 8 solution) \\
2936 \sudoku{cs5.ans}
2937 \end{center}\end{minipage}
2938
2939 \renewcommand*{\puzzlefile}{cs6.ans}
2940 \writepuzzle%
2941 {312654978}{645987312}{978321645}%
2942 {756198423}{423765189}{189432756}%
2943 {534876291}{291543867}{867219534}%
2944 [CreateSudoku 6 (level 4) solution]
2945 \renewcommand*{\puzzlefile}{sh5.ans}
2946 \writepuzzle%
2947 {269175438}{471832965}{583694217}%
2948 {715426389}{894317526}{326958741}%
2949 {942561873}{138749652}{657283194}%
2950 [SSBL hard 5 solution]
2951 \vfill
2952 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2953 CS6 (level 4 solution) \\
2954 \sudoku{cs6.ans}
2955 \end{center}\end{minipage}
2956 \hfill
2957 \begin{minipage}{0.47\linewidth}\begin{center}
2958 SH5 (hard solution) \\
2959 \sudoku{sh5.ans}
2960 \end{center}\end{minipage}
2961
2962 \renewcommand*{\puzzlefile}{sh10.ans}
2963 \writepuzzle%
2964 {437921865}{156837492}{289456137}%
2965 {598163274}{374295681}{621748953}%
2966 {713682549}{845319726}{962574318}%
2967 [SSBL hard 10 solution]
2968 \renewcommand*{\puzzlefile}{sh15.ans}
2969 \writepuzzle%
2970 {216935847}{478261359}{395478261}%
2971 {761392584}{952846713}{843157926}%
2972 {124783695}{539624178}{687519432}%
2973 [SSBL hard 15 solution]

```

```

2974 \vfill
2975 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2976 SH10 (hard solution) \\
2977 \sudoku{sh10.ans}
2978 \end{center}\end{minipage}
2979 \hfill
2980 \begin{minipage}{0.47\linewidth}\begin{center}
2981 SH15 (hard solution) \\
2982 \sudoku{sh15.ans}
2983 \end{center}\end{minipage}
2984
2985 \renewcommand*{\puzzlefile}{sh20.ans}
2986 \writepuzzle%
2987 {468271935}{925438671}{713695284}%
2988 {154762893}{832519467}{679384512}%
2989 {587943126}{296157348}{341826759}%
2990 [SSBL hard 20 solution]
2991 \renewcommand*{\puzzlefile}{sh25.ans}
2992 \writepuzzle%
2993 {439561728}{167238459}{852497613}%
2994 {286953174}{794816532}{513724986}%
2995 {348179265}{621345897}{975682341}%
2996 [SSBL hard 25 solution]
2997 \vfill
2998 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
2999 SH20 (hard solution) \\
3000 \sudoku{sh20.ans}
3001 \end{center}\end{minipage}
3002 \hfill
3003 \begin{minipage}{0.47\linewidth}\begin{center}
3004 SH25 (hard solution) \\
3005 \sudoku{sh25.ans}
3006 \end{center}\end{minipage}
3007
3008 \renewcommand*{\puzzlefile}{sh30.ans}
3009 \writepuzzle%
3010 {461258793}{253917846}{789463152}%
3011 {346789521}{572146389}{918325674}%
3012 {894671235}{635892417}{127534968}%
3013 [SSBL hard 30 solution]
3014 \renewcommand*{\puzzlefile}{tt100.ans}
3015 \writepuzzle%
3016 {513896427}{742315698}{968742531}%
3017 {154289376}{376154289}{289637154}%
3018 {891563742}{425971863}{637428915}%
3019 [DT2 tough 100 solution]
3020 \vfill
3021 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
3022 SH30 (hard solution) \\
3023 \sudoku{sh30.ans}

```

```

3024 \end{center}\end{minipage}
3025 \hfill
3026 \begin{minipage}{0.47\linewidth}\begin{center}
3027 TT100 (tough solution) \\
3028 \sudoku{tt100.ans}
3029 \end{center}\end{minipage}
3030
3031 \renewcommand*{\puzzlefile}{tt105.ans}
3032 \writepuzzle%
3033 {925378146}{681254793}{734916582}%
3034 {256739418}{178462359}{349581267}%
3035 {597843621}{462197835}{813625974}%
3036 [DT2 tough 105 solution]
3037 \renewcommand*{\puzzlefile}{tt110.ans}
3038 \writepuzzle%
3039 {379412586}{648539217}{125687349}%
3040 {936825174}{214763958}{857194632}%
3041 {492358761}{581276493}{763941825}%
3042 [DT2 tough 110 solution]
3043 \vfill
3044 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
3045 TT105 (tough solution) \\
3046 \sudoku{tt105.ans}
3047 \end{center}\end{minipage}
3048 \hfill
3049 \begin{minipage}{0.47\linewidth}\begin{center}
3050 TT110 (tough solution) \\
3051 \sudoku{tt110.ans}
3052 \end{center}\end{minipage}
3053
3054 \renewcommand*{\puzzlefile}{tt115.ans}
3055 \writepuzzle%
3056 {432758196}{196243587}{758169423}%
3057 {945682371}{263571849}{871934265}%
3058 {329416758}{584397612}{617825934}%
3059 [DT2 tough 115 solution]
3060 \renewcommand*{\puzzlefile}{tt120.ans}
3061 \writepuzzle%
3062 {968215734}{143679852}{752384619}%
3063 {496523178}{537198246}{281746395}%
3064 {875961423}{314852967}{629437581}%
3065 [DT2 tough 120 solution]
3066 \vfill
3067 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
3068 TT115 (tough solution) \\
3069 \sudoku{tt115.ans}
3070 \end{center}\end{minipage}
3071 \hfill
3072 \begin{minipage}{0.47\linewidth}\begin{center}
3073 TT120 (tough solution) \\

```

```

3074 \sudoku{tt120.ans}
3075 \end{center}\end{minipage}
3076
3077 \renewcommand*{\puzzlefile}{st1130.ans}
3078 \writepuzzle%
3079 {819356247}{576924813}{342817596}%
3080 {781563924}{965248371}{423179685}%
3081 {658791432}{297435168}{134682759}
3082 [Seattle Times 2005/11/30 3 star solution]
3083 \renewcommand*{\puzzlefile}{st1207.ans}
3084 \writepuzzle%
3085 {217483956}{834695712}{659172384}%
3086 {798231645}{326954178}{145768239}%
3087 {582319467}{971846523}{463527891}%
3088 [Seattle Times 2005/12/07 3 star solution]
3089 \vfill
3090 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
3091 ST1130 (3 star solution) \\
3092 \sudoku{st1130.ans}
3093 \end{center}\end{minipage}
3094 \hfill
3095 \begin{minipage}{0.47\linewidth}\begin{center}
3096 ST1207 (3 star solution) \\
3097 \sudoku{st1207.ans}
3098 \end{center}\end{minipage}
3099
3100 \renewcommand*{\puzzlefile}{st1209.ans}
3101 \writepuzzle%
3102 {839652147}{246781359}{751943268}%
3103 {167295483}{592438716}{483167925}%
3104 {628319574}{374526891}{915874632}%
3105 [Seattle Times 2005/12/09 5 star solution]
3106 \renewcommand*{\puzzlefile}{st1221.ans}
3107 \writepuzzle%
3108 {517863924}{436925178}{298417563}%
3109 {174389652}{683251749}{952674381}%
3110 {745132896}{369548217}{821796435}%
3111 [Seattle Times 2005/12/21 3 star solution]
3112 \vfill
3113 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
3114 ST1209 (5 star solution) \\
3115 \sudoku{st1209.ans}
3116 \end{center}\end{minipage}
3117 \hfill
3118 \begin{minipage}{0.47\linewidth}\begin{center}
3119 ST1221 (3 star solution) \\
3120 \sudoku{st1221.ans}
3121 \end{center}\end{minipage}
3122
3123 \renewcommand*{\puzzlefile}{st1228.ans}

```

```

3124 \writepuzzle%
3125 {673189524}{812654379}{594273186}%
3126 {265791843}{341865792}{987342651}%
3127 {795416238}{138927465}{426538917}%
3128 [Seattle Times 2005/12/28 3 star solution]
3129 \renewcommand*{\puzzlefile}{st0126.ans}
3130 \writepuzzle%
3131 {253698471}{197435286}{864127953}%
3132 {416573829}{785912634}{329864517}%
3133 {972356148}{638241795}{541789362}%
3134 [Seattle Times 2006/01/26 4 star solution]
3135 \vfill
3136 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
3137 ST1228 (3 star solution) \\
3138 \sudoku{st1228.ans}
3139 \end{center}\end{minipage}
3140 \hfill
3141 \begin{minipage}{0.47\linewidth}\begin{center}
3142 ST0126 (4 star solution) \\
3143 \sudoku{st0126.ans}
3144 \end{center}\end{minipage}
3145
3146 \renewcommand*{\puzzlefile}{st0127.ans}
3147 \writepuzzle%
3148 {253816479}{467395812}{918427536}%
3149 {641752398}{892163745}{375948621}%
3150 {189534267}{736289154}{524671983}%
3151 [Seattle Times 2006/01/27 5 star solution]
3152 \renewcommand*{\puzzlefile}{st0204.ans}
3153 \writepuzzle%
3154 {153427896}{742986513}{869513247}%
3155 {375142968}{428369751}{916875432}%
3156 {597321684}{681754329}{234698175}%
3157 [Seattle Times 2006/02/04 6 star solution]
3158 \vfill
3159 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
3160 ST0127 (5 star solution) \\
3161 \sudoku{st0127.ans}
3162 \end{center}\end{minipage}
3163 \hfill
3164 \begin{minipage}{0.47\linewidth}\begin{center}
3165 ST0204 (6 star solution) \\
3166 \sudoku{st0204.ans}
3167 \end{center}\end{minipage}
3168
3169 \renewcommand*{\puzzlefile}{st0208.ans}
3170 \writepuzzle%
3171 {213985647}{589674213}{647213859}%
3172 {928751364}{361429578}{754836192}%
3173 {875192436}{432567981}{196348725}%

```

```

3174 [Seattle Times 2005/02/08 3 star solution]
3175 \renewcommand*{\puzzlefile}{st0209.ans}
3176 \writepuzzle%
3177 {639587421}{752341698}{841296753}%
3178 {584769312}{976123584}{123458976}%
3179 {297834165}{318675249}{465912837}%
3180 [Seattle Times 2005/02/09 4 star solution]
3181 \vfill
3182 \noindent\begin{minipage}{0.47\linewidth}\begin{center}
3183 ST0208 (3 star solution) \\
3184 \sudoku{st0208.ans}
3185 \end{center}\end{minipage}
3186 \hfill
3187 \begin{minipage}{0.47\linewidth}\begin{center}
3188 ST0209 (4 star solution) \\
3189 \sudoku{st0209.ans}
3190 \end{center}\end{minipage}
3191
3192
3193 %%%%%%%%%%%%%%% bibliography
3194 \cleardoublepage
3195
3196 \bibliographystyle{alpha}
3197 \begin{group}
3198 \raggedright
3199 \begin{thebibliography}{GMSN94A}
3200
3201 \bibitem[Mep05]{DT2}
3202 Michael Mepham.
3203 \emph{The Daily Telegraph Sudoku 2}.
3204 \newblock Pan Books, 2005.
3205 \newblock ISBN 0-330-44203-3.
3206
3207 \bibitem[Wil06]{SUDOKUBUNDLE}
3208 Peter Wilson.
3209 \emph{The sudoku bundle for displaying, solving and generating
3210 Sudoku puzzles}.
3211 \newblock February 2006.
3212 \newblock (Available from CTAN in \url{macros/latex/contrib/sudokubundle}).
3213
3214 \bibitem[SOL]{SOL}
3215 \emph{Sudoku Online: Home of the Sudokulist}.
3216 \newblock \url{http://www.sudoku.org.uk}
3217
3218 \bibitem[SSBL]{SSBL}
3219 \emph{Sudoku Solver ... by logic}.
3220 \newblock \url{http://www.sudokusolver.co.uk}
3221
3222 \end{thebibliography}
3223 \endgroup

```

```

3224
3225 \end{document}
3226
3227 \end{examples}

```

References

- [Abr05] Paul Abraham. *The sudoku package*. June 2005. (Available from CTAN in `macros/latex/contrib/sudoku`)
- [Ars95] Donald Arseneau. *Generating random numbers in T_EX*. 1995. (Available from CTAN in `macros/generic/random.tex`)
- [MG04] Frank Mittelbach and Michel Goossens. *The LaTeX Companion*. Second edition. Addison-Wesley Publishing Company, 2004. ISBN 0-201-36299-6.
- [PJ05] T_EX Users Group. Distractions: Sudoku. *The PracTeX Journal*. 2005-4. <http://tug.org/pracjourn2005-4/distract>
- [SOL] *Sudoku Online: Home of the Sudokulist*. <http://www.sudoku.org.uk>
- [SSBL] *Sudoku Solver ... by logic*. <http://www.sudokusolver.co.uk>

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.

A	B	<u>13</u> , 1518, 1848, 2544
<code>\acluenotdeletedfalse</code> 1768	<code>\baselineskip</code> 18, 1518, 1848, 2544	<code>\centering</code> 1833
<code>\acluenotdeletedtrue</code> 1761	<code>\bibitem</code> 3201, 3207, 3214, 3218	<code>\changedfalse</code> . 684, 752
<code>\adds@dcclues</code> <u>29</u> , 91	<code>\bibliographystyle</code> 3196	<code>\changedtrue</code> 578, 605, 633, 647, 661, 739
<code>\anychangefalse</code> ... 702	<code>\boxctr</code> <u>129</u> , 554, 555, 559, 563, 928,	<code>\checkboxes</code> . 1365, <u>1398</u>
<code>\anychangetrue</code> 693, 704, 712	930, 931, 949, 953, 1246, 1248,	<code>\checkcols</code> . 1364, <u>1382</u>
<code>\asetctr</code> . <u>141</u> , 1078, 1079, 1091, 1092, 1186, 1187, 1199, 1200, 1298, 1299, 1315, 1316	1249, 1264, 1267, 1401, 1403, 1404, 1417, 1418, 1420	<code>\checkkeepon</code> .. 716, <u>725</u>
<code>\author</code> 1800	C	<code>\checkrows</code> . 1363, <u>1367</u>
	<code>\cellsize</code> 4,	<code>\checksetforpair</code> 1092, <u>1104</u> , 1200, 1316
		<code>\checkssimplereductions</code> 743, <u>747</u>
		<code>\checksolution</code> ... <u>1362</u>
		<code>\cleardoublepage</code> 1845, 2539, 3194

- `\clearpage` 1214, 1231, 1320, 1330, 1348
 .. 2071, 2280, 2537
`\closein` 10, 44, 496, 515
`\closeout`
 101, 118, 1460,
 1469, 1562,
 1573, 1780, 1789
`\cluefont` 4,
 13, 1517, 1847, 2543
`\commentary`
 . 127, 582, 609,
 637, 651, 665,
 735, 759, 856,
 918, 998, 1009,
 1021, 1033, 1530
`\createsudsets` ... 167
`\currfile`
 . 7, 1534, 1644,
 1740, 1743, 1746
- D**
- `\date` 1801
`\deleteaclue` 1745, 1759
`\deleteboxpairedigits`
 1321, 1328
`\deletecolpairedigits`
 1205, 1212
`\deletenumfromset` .
 336,
 576, 603, 631,
 645, 659, 1141,
 1147, 1225,
 1231, 1342, 1348
`\deleterowpairedigits`
 1097, 1128
`\difficultyctr` 6, 141,
 186, 680, 681,
 699, 708, 719,
 722, 736, 737,
 761, 762, 1035, 1036
`\digitictr` 151, 1096,
 1105, 1111,
 1119, 1130,
 1141, 1204,
 1214, 1225,
 1320, 1330, 1342
`\digitiictr` 151, 1096,
 1106, 1112,
 1120, 1130,
 1147, 1204,
- 1214, 1231,
 1320, 1330, 1348
`\displaystatus`
 .. 404, 1430, 1440
`\dos@dcols` 41, 46
`\dos@dinitialcols` .
 512, 517
- E**
- `\elimclues` . 1554, 1641
`\elimcluesonebyone` .
 1645, 1732
`\elimcross`
 . 7, 1661, 1689,
 1716, 1721, 1727
`\elimcrossandex` ...
 7, 1715,
 1729, 1837, 1839
`\elimcrossandexandnines`
 . 7, 1720, 1730,
 1836, 1838, 1841
`\elimcrossandnines` .
 7, 1688, 1728
`\elimex`
 7, 1700, 1717, 1722
`\elimnum` .. 7, 1677, 1723
`\elimseventeen`
 7, 1650, 1726, 1840
`\emph` 1813–1815, 3203,
 3209, 3215, 3219
- F**
- `\findboxpair` 1301, 1307
`\findcolpair` 1188, 1194
`\findrowpair` 1080, 1086
`\firsts@dcluefalse` .
 66, 530
`\firsts@dcluetrue` .
 31, 36, 66, 504, 510
`\fixentry` 521, 526
`\footnotesize` 1429, 1439
- G**
- `\g@addto@macro`
 .. 1482, 1486–1495
`\gatherline`
 .. 1465, 1471, 1785
`\gencommentary`
 .. 1530, 1577, 1766
`\generategrid` .. 6, 1546
- `\genfile`
 7, 1534, 1557, 1647
`\getloner`
 . 787, 815, 872, 944
`\getnthboxcell` 1271,
 1297, 1314, 1341
`\getproblem`
 .. 6, 492, 1510,
 1549, 1551,
 1646, 1740, 1743
`\gets@dclue` 54, 66
`\gettwo` 59
- H**
- `\halfs@dcell` 13, 83
`\hideprogress` 1445
`\hideprogress` 1444
`\hline` 406,
 415, 424, 433,
 442, 451, 460,
 469, 478, 487,
 1834, 1835, 1841
`\Huge` 21
- I**
- `\ifacluenotdeleted` 1759
`\ifanychange`
 161, 706, 729
`\ifanychanged` 156
`\ifchanged` 156, 741, 748
`\iffirstchar` 156
`\iffirsts@dclue` ...
 66, 156, 528
`\ifkeepon` 156, 696
`\iflonerchanged` ...
 156, 703,
 767, 769, 791,
 876, 948, 1001, 1013
`\ifnotgotthechar` .. 156
`\ifodd` 209, 216,
 223, 230, 237,
 244, 251, 258,
 265, 349, 355,
 361, 367, 373,
 379, 385, 391, 396
`\ifpairchanged` 156,
 711, 1040, 1042,
 1060, 1169, 1263
`\ifsetchanged` . 156,
 577, 604, 632,

- 646, 660, 1143,
1149, 1227,
1233, 1344, 1350
`\ifstilldigits` 156
`\immediate`
. 101–111, 115,
116, 118, 496,
497, 515, 1460,
1461, 1466,
1469, 1562–
1573, 1780,
1781, 1786, 1789
`\initialelimination`
. 7, 1642, 1726
`\initialisesuddata` .
. 182, 494
`\initialsoln` . . 498, 502
`\istchar`
59, 76, 78, 534, 536
- K**
- `\keeponfalse` . . 727, 730
`\keepontrue` 694
`\keepreducing`
. 6, 692, 1512, 1775
`\keepreducingcells` .
. 697, 733
- L**
- `\LARGE` 1847
`\line` 86, 87, 89, 90
`\linethickness` 88
`\linewidth` 1426, 1436,
1514, 1860,
1872, 1884,
1896, 1908,
1920, 1932,
1944, 1956,
1968, 1980,
1992, 2004,
2016, 2028,
2040, 2052,
2064, 2081,
2093, 2105,
2117, 2129,
2141, 2153,
2165, 2177,
2189, 2201,
2213, 2225,
2237, 2249,
- 2261, 2273,
2290, 2302,
2314, 2326,
2338, 2350,
2362, 2374,
2386, 2398,
2410, 2422,
2434, 2446,
2458, 2470,
2482, 2494,
2506, 2518,
2530, 2559,
2564, 2582,
2587, 2605,
2610, 2628,
2633, 2651,
2656, 2674,
2679, 2697,
2702, 2720,
2725, 2743,
2748, 2768,
2773, 2791,
2796, 2814,
2819, 2837,
2842, 2860,
2865, 2883,
2888, 2906,
2911, 2929,
2934, 2952,
2957, 2975,
2980, 2998,
3003, 3021,
3026, 3044,
3049, 3067,
3072, 3090,
3095, 3113,
3118, 3136,
3141, 3159,
3164, 3182, 3187
`\lonecellctr`
. 141, 850–
853, 912–915,
992–995, 1003–
1006, 1015–1018
`\lonerchangedfalse` . 760
`\lonerchangedtrue` .
. 854,
916, 996, 1007, 1019
`\loop` . . . 33, 49, 175,
188, 507, 519,
- 574, 601, 628,
685, 695, 740,
781, 806, 821,
848, 867, 891,
910, 929, 963,
989, 1053, 1076,
1089, 1113,
1135, 1163,
1184, 1197,
1219, 1247,
1295, 1312,
1335, 1370,
1386, 1402,
1463, 1478,
1579, 1613,
1630, 1653,
1664, 1670,
1679, 1691,
1702, 1708,
1737, 1762, 1783
`\Lpack` 1798,
1816, 1821, 1829
- M**
- `\makebox` 54
`\maketitle` 1804
`\markboth` 1850,
2072, 2281, 2541
`\maxrangectr`
. 141, 1073,
1074, 1077,
1181, 1182,
1185, 1293,
1294, 1296,
1576, 1577,
1580, 1612,
1614, 1629,
1631, 1652, 1654
`\multiply` 38, 52, 211,
213, 218, 220,
225, 227, 232,
234, 239, 241,
246, 248, 253,
255, 260, 262,
267, 269, 294,
299, 304, 309,
314, 319, 324,
328, 551, 555,
558, 784, 1056, 1373
`\multiput` 86, 87, 89, 90

- N**
- \newblock 3204,
3205, 3211,
3212, 3216, 3220
 - \newcount . . . 24–27,
130–154, 169, 1732
 - \newif 67, 157–165, 1759
 - \newknt 167
 - \newread 5
 - \newwrite 6
 - \nextrandom 1542
 - \normalsize . 1517, 2543
 - \nowt 59
 - \numcluesctr
. . . 6, 151, 187, 538
 - \numdigitsctr . 151,
1093, 1107,
1110, 1117,
1118, 1201, 1317
 - \numlistctr 141, 579,
580, 583, 584,
606, 607, 610,
611, 634, 635,
638, 639, 648,
649, 652, 653,
662, 663, 666, 667
 - \numofnuminset
. 279, 823,
850, 912, 992,
1003, 1015, 1115
- O**
- \oldcommentary . . . 1530
 - \openin 11, 497
 - \openout 102,
1461, 1563, 1781
- P**
- \pagestyle
. . . 1851, 2538, 2540
 - \pairchangedfalse .
. 709, 1034,
1132, 1216, 1332
 - \pairchangedtrue . .
. 1145,
1151, 1229,
1235, 1346, 1352
 - \prevfile . . 7, 1534,
1549, 1563,
1643, 1646, 1744
 - \printsudresults . 1546
2922, 2939,
2945, 2962,
2968, 2985,
2991, 3008,
3014, 3031,
3037, 3054,
3060, 3077,
3083, 3100,
3106, 3123,
3129, 3146,
3152, 3169, 3175
 - \ProvidesPackage . .
. 3, 124, 1526
 - \put 54
 - \puzzlefile
. . . 4, 99, 1853,
1865, 1877,
1889, 1901,
1913, 1925,
1937, 1949,
1961, 1973,
1985, 1997,
2009, 2021,
2033, 2045,
2057, 2074,
2086, 2098,
2110, 2122,
2134, 2146,
2158, 2170,
2182, 2194,
2206, 2218,
2230, 2242,
2254, 2266,
2283, 2295,
2307, 2319,
2331, 2343,
2355, 2367,
2379, 2391,
2403, 2415,
2427, 2439,
2451, 2463,
2475, 2487,
2499, 2511,
2523, 2546,
2552, 2569,
2575, 2592,
2598, 2615,
2638, 2661,
2667, 2684,
2690, 2707,
2713, 2730,
2736, 2755,
2761, 2778,
2784, 2801,
2807, 2824,
2830, 2847,
2853, 2870,
2876, 2893,
2899, 2916,
- R**
- \raggedbottom 1802
 - \raggedright 3198
 - \randomi 1538
 - \read 35, 509
 - \reads@game 8, 96
 - \reduceaboxpair
. 1262, 1288
 - \reduceacell 543, 687
 - \reduceacolpair
. 1168, 1176
 - \reduceallcells . 6,
677, 742, 1511, 1774
 - \reducelonrowpair
. 1059, 1067
 - \reducebox 563, 621
 - \reduceboxloners
. 770, 925
 - \reduceboxpairs
. 1043, 1243
 - \reducecol 562, 594
 - \reducecolloners
. 768, 863
 - \reducecolpairs
. 1041, 1159
 - \reducedctr
. 141, 576, 579,
586, 603, 606,
613, 631, 634,
641, 645, 648,
655, 659, 662, 669
 - \reducelonerboxcell
. 947, 981
 - \reducelonercolcell
. 875, 900
 - \reducelonerrowcell . 838
 - \reducelonerrowcell
. 790, 841

- `\reduceloners` . 700, [757](#)
 - `\reducepairs` . 710, [1031](#)
 - `\reducerow` ... 561, [567](#)
 - `\reducerowloners` ..
..... 766, [777](#)
 - `\reducerowpairs` ...
..... 1039, [1049](#)
 - `\repeat` 43,
56, 179, 192,
514, 523, 590,
617, 673, 689,
717, 744, 797,
812, 829, 860,
882, 897, 922,
954, 978, 1026,
1065, 1083,
1101, 1125,
1156, 1174,
1191, 1209,
1240, 1268,
1304, 1325,
1357, 1380,
1395, 1421,
1468, 1499,
1604, 1621,
1638, 1658,
1668, 1674,
1685, 1697,
1706, 1712,
1756, 1770, 1788
 - `\RequirePackage` ...
..... 125, 1528
 - `\restchars` . [59](#), 74, 532
- S**
- `\s@dcolpos` [23](#), 51–54,
552, 562, 1610,
1616, 1617, 1619
 - `\s@dfont` 20, 54
 - `\s@dfstop` .. [63](#), 76, 534
 - `\s@dgame` [81](#), 97
 - `\s@dline` 35, 71, 509, 529
 - `\s@dncol` [23](#),
40, 50, 51, 55,
511, 520, 522,
570, 571, 575,
576, 583, 584,
586, 589, 624,
625, 629, 630,
672, 802, 803,
 - 807–809, 811,
842, 843, 846,
849, 850, 853,
857, 859, 866,
868, 869, 877,
881, 959, 960,
964, 965, 977,
985, 986, 990,
991, 999, 1010,
1022, 1025,
1070, 1071,
1077, 1078,
1082, 1087,
1095, 1129,
1137, 1162,
1164, 1165,
1170, 1173,
1385, 1387,
1388, 1391,
1392, 1394,
1475, 1476,
1479, 1480,
1498, 1609,
1615, 1616, 1618
 - `\s@dnrow` [23](#),
32, 34, 37, 42,
505, 508, 513,
597, 598, 602,
603, 610, 611,
613, 616, 630,
631, 638, 639,
641, 644, 645,
652, 653, 655,
658, 659, 666,
667, 669, 780,
782, 783, 792,
796, 887, 888,
892–894, 896,
904, 905, 908,
911, 912, 915,
919, 921, 965–
967, 969–971,
973–975, 991,
992, 995, 1002,
1003, 1006,
1014, 1015,
1018, 1052,
1054, 1055,
1061, 1064,
1178, 1179,
 - 1185, 1186,
1190, 1195,
1203, 1213,
1221, 1300,
1319, 1329,
1337, 1369,
1371, 1372,
1376, 1377,
1379, 1462,
1464, 1465,
1467, 1626,
1632, 1633,
1635, 1782,
1784, 1785, 1787
 - `\s@dread`
5, 10, 11, 35, 44,
496, 497, 509, 515
 - `\s@drowpos`
. [23](#), 37–39, 54,
553, 561, 1627,
1633, 1634, 1636
 - `\s@dttempcnta` .. [141](#),
407–487, 547,
561–563, 823–
825, 830, 831,
833, 834, 1115,
1116, 1142,
1144, 1148,
1150, 1226,
1228, 1232,
1234, 1273,
1276–1283,
1285, 1297,
1298, 1343,
1345, 1349,
1351, 1480,
1481, 1484,
1485, 1615,
1617, 1632,
1634, 1655,
1656, 1764–1767
 - `\s@dttempcntb` [141](#), 787,
788, 790, 872,
873, 875, 944,
945, 947, 1314,
1315, 1341,
1343, 1345,
1349, 1351,
1581–1583,

- 1736, 1738,
1747, 1750, 1754
- `\s@dttemploopcnta` [141](#),
820, 822, 823,
825, 826, 828,
830, 1109, 1114,
1115, 1119–
1121, 1124,
1133, 1134,
1136, 1137,
1139, 1142,
1144, 1148,
1150, 1155,
1217, 1218,
1220, 1221,
1223, 1226,
1228, 1232,
1234, 1239,
1290, 1291,
1296, 1297,
1300, 1303,
1310, 1333,
1334, 1336,
1337, 1339,
1341, 1356,
1578, 1580,
1582, 1603,
1611, 1614,
1620, 1628,
1631, 1637,
1651, 1654,
1657, 1663,
1665–1667,
1669, 1671–
1673, 1678,
1680–1682,
1684, 1690,
1692–1694,
1696, 1701,
1703–1705,
1707, 1709–1711
- `\s@dwrite` [5](#),
101–111, 115,
116, 118, 1460,
1461, 1466,
1469, 1562–
1573, 1780,
1781, 1786, 1789
- `\s@lcnta` [129](#),
173, 176–178,
- 184, 189–191,
506, 536, 540,
546, 547, 549,
552, 556, 683,
686, 688, 783,
784, 790, 802,
869, 875, 887,
932–940, 947,
959, 985, 1055,
1056, 1059,
1094, 1128,
1133, 1165,
1168, 1202,
1212, 1217,
1250–1258,
1262, 1318,
1372, 1373,
1388, 1405–1413
- `\secondctr` [141](#), 1087,
1088, 1090,
1091, 1095,
1100, 1129,
1139, 1195,
1196, 1198,
1199, 1203,
1208, 1213,
1223, 1310,
1311, 1313,
1314, 1319,
1324, 1329, 1339
- `\setchangedfalse` .. 341
- `\setchangedtrue` ...
..... 351, 357,
363, 369, 375,
381, 387, 393, 398
- `\setrannum` ... 1576,
1581, 1655, 1764
- `\setsudrandom` .. 7, [1538](#)
- `\settonum` [197](#),
407–487, 579,
606, 634, 648, 662
- `\settonumcnt` 132, 202,
205, 206, 209,
215, 216, 222,
223, 229, 230,
236, 237, 243,
244, 250, 251,
257, 258, 264, 265
- `\showprogress` [1444](#)
- `\showprogress` 1449, 1453
- `\solcnt` [129](#),
174, 185, 495,
499, 537, 581,
582, 608, 609,
636, 637, 650,
651, 664, 665,
698, 707, 718,
721, 726, 751,
764, 855, 856,
917, 918, 997,
998, 1008, 1009,
1020, 1021,
1038, 1428,
1438, 1742, 1749
- `\splitoff`
59, 71, 74, 529, 532
- `\sudaline`
.. 1466, [1471](#), 1786
- `\sudoksolve` 1503
- `\sudoku` . 4, [94](#), 1508,
1519, 1862,
1874, 1886,
1898, 1910,
1922, 1934,
1946, 1958,
1970, 1982,
1994, 2006,
2018, 2030,
2042, 2054,
2066, 2083,
2095, 2107,
2119, 2131,
2143, 2155,
2167, 2179,
2191, 2203,
2215, 2227,
2239, 2251,
2263, 2275,
2292, 2304,
2316, 2328,
2340, 2352,
2364, 2376,
2388, 2400,
2412, 2424,
2436, 2448,
2460, 2472,
2484, 2496,
2508, 2520,
2532, 2561,
2566, 2584,

- 2589, 2607,
2612, 2630,
2635, 2653,
2658, 2676,
2681, 2699,
2704, 2722,
2727, 2745,
2750, 2770,
2775, 2793,
2798, 2816,
2821, 2839,
2844, 2862,
2867, 2885,
2890, 2908,
2913, 2931,
2936, 2954,
2959, 2977,
2982, 3000,
3005, 3023,
3028, 3046,
3051, 3069,
3074, 3092,
3097, 3115,
3120, 3138,
3143, 3161,
3166, 3184, 3189
- `\sudokusolve`
 5, 1505, 1557
- `\sudpuzznewline` .. 4, 99
- `\sudsolnfile`
 6, 1457, 1519
- `\sumboxsets`
 942, 957, 1260, 1415
- `\sumcolsets`
 870, 885, 1166, 1389
- `\sumctr` ... 129, 779,
 786, 787, 805,
 809, 865, 871,
 872, 890, 894,
 927, 943, 944,
 962, 967, 971,
 975, 988, 1051,
 1058, 1075,
 1161, 1167,
 1183, 1245,
 1261, 1368,
 1375, 1384,
 1390, 1400, 1416
- `\sumrowsets`
 785, 799, 1057, 1374
- `\swapcolpair`
 .. 1584–1592, 1624
- `\swaprowpair`
 .. 1593–1601, 1607
- `\swaps` 1553, 1575
- T**
- `\tempcnty` . 129, 203,
 206, 207, 212,
 219, 226, 233,
 240, 247, 254,
 261, 268, 271–
 273, 276, 549–
 551, 554, 556–559
- `\tempcntz`
 . 129, 210–212,
 217–219, 224–
 226, 231–233,
 238–240, 245–
 247, 252–254,
 259–261, 266–
 268, 551–553, 557
- `\tenscnt` 129,
 204, 211, 213,
 218, 220, 225,
 227, 232, 234,
 239, 241, 246,
 248, 253, 255,
 260, 262, 267, 269
- `\textsc` 1798
- `\thicklines` 88
- `\thinlines` 85
- `\thr@@` 555, 558, 626
- `\title` 1799
- `\tmpsetansctr`
 . 129, 284, 286,
 290, 292, 293,
 295, 297, 298,
 300, 302, 303,
 305, 307, 308,
 310, 312, 313,
 315, 317, 318,
 320, 322, 323,
 325, 327, 329,
 331, 334, 343,
 350, 356, 362,
 368, 374, 380,
 386, 392, 397, 402
- `\tmpsetctr` ... 129,
 283–285, 293–
- 295, 298–300,
303–305, 308–
310, 313–315,
318–320, 323–
325, 327–329,
342, 344, 348,
349, 354, 355,
360, 361, 366,
367, 372, 373,
378, 379, 384,
385, 390, 391, 396
- `\toomanyloops` 1732
- `\toprangectr` 129, 571,
 572, 575, 598,
 599, 602, 625,
 626, 629, 803,
 804, 807, 843,
 844, 849, 857,
 888, 889, 892,
 905, 906, 911,
 919, 960, 961,
 964, 986, 987,
 990, 999, 1010,
 1022, 1071–
 1073, 1090,
 1134, 1136,
 1179–1181,
 1198, 1218,
 1220, 1291–
 1293, 1313,
 1334, 1336,
 1476, 1477, 1479
- `\trysolution` 1741, 1773
- `\typelonestatus` ...
 .. 773, 1448, 1451
- `\typelonestatusX` ..
 1434, 1451
- `\typesimplestatus` .
 .. 749, 1447, 1450
- `\typesimplestatusX` .
 1424, 1450
- U**
- `\unitlength` 83
- `\url` ... 3212, 3216, 3220
- `\useknt` 167, 190, 407–
 487, 536, 546,
 547, 576, 584,
 586, 603, 611,
 613, 631, 639,

- 641, 645, 653,
655, 659, 667,
669, 808, 809,
850, 853, 893,
894, 912, 915,
966, 967, 970,
971, 974, 975,
992, 995, 1003,
1006, 1015,
1018, 1078,
1091, 1142,
1144, 1148,
1150, 1186,
1199, 1226,
1228, 1232,
1234, 1298,
1315, 1343,
1345, 1349,
1351, 1480,
1615–1617,
1632–1634,
1656, 1666,
1672, 1681,
1682, 1693,
1694, 1704,
1710, 1765, 1767
- V**
- `\verb` 1835–1841
- W**
- `\write` 103–111,
115, 116, 1466,
1564–1572, 1786
`\writegame` 6, 1457, 1513
`\writepuzzle`
. . . 4, 99, 1854,
1866, 1878,
1890, 1902,
1914, 1926,
1938, 1950,
1962, 1974,
1986, 1998,
2010, 2022,
2034, 2046,
2058, 2075,
2087, 2099,
2111, 2123,
2135, 2147,
2159, 2171,
2183, 2195,
2207, 2219,
2231, 2243,
2255, 2267,
2284, 2296,
2308, 2320,
2332, 2344,
2356, 2368,
2380, 2392,
2404, 2416,
2428, 2440,
2452, 2464,
2476, 2488,
2500, 2512,
2524, 2547,
2553, 2570,
2576, 2593,
2599, 2616,
2622, 2639,
2645, 2662,
2668, 2685,
2691, 2708,
2714, 2731,
2737, 2756,
2762, 2779,
2785, 2802,
2808, 2825,
2831, 2848,
2854, 2871,
2877, 2894,
2900, 2917,
2923, 2940,
2946, 2963,
2969, 2986,
2992, 3009,
3015, 3032,
3038, 3055,
3061, 3078,
3084, 3101,
3107, 3124,
3130, 3147,
3153, 3170, 3176
`\writes@dpuzzend` . . 99
`\writestartgrid` . . .
. 1548, 1560
`\writestate` . . . 1643,
1644, 1647,
1744, 1746, 1778