*Theory and Practice of Logic Programming: LaTeX Guidelines for authors**

CAMBRIDGE AUTHOR1
Cambridge University Press

CAMBRIDGE AUTHOR2
Cambridge University Press

Abstract

This guide is for authors who are preparing papers for the *Theory and Practice of Logic Programming* journal using the LaTeX document preparation system and the CUP TLP style file.

KEYWORDS: Key1, Key2, Key3, Key4

1 Introduction

The layout design for the *Theory and Practice of Logic Programming* journal has been implemented as a LaTeX style file. The TLP style file is based on the ARTICLE style as discussed in the LaTeX manual. Commands which differ from the standard LaTeX interface, or which are provided in addition to the standard interface, are explained in this guide. This guide is not a substitute for the LaTeX manual itself.

1.1 Introduction to LaTeX

The LaTeX document preparation system is a special version of the TeX typesetting program. LaTeX adds to TeX a collection of commands which simplify typesetting by allowing the author to concentrate on the logical structure of the document rather than its visual layout.

LaTeX provides a consistent and comprehensive document preparation interface. There are simple-to-use commands for generating a table of contents, lists of figures and/or tables, and indexes. LaTeX can automatically number list entries, equations, figures, tables, and footnotes, as well as parts, chapters, sections and subsections. Using this numbering system, bibliographic citations, page references and cross references to any other numbered entity (*e.g.* chapter, section, equation, figure, list entry) are quite straightforward.

1.2 The TLP document class

The use of document class allows a simple change of style (or style option) to transform the appearance of your document. The CUP TLP class file preserves the standard LaTeX[1] interface such that any document which can be produced using the standard LaTeX ARTICLE style can also be produced with the TLP style. However, the fonts (sizes) and measure of text is slightly different from that for ARTICLE, therefore line breaks will change and it is possible that equations may need re-setting.

2 Additional facilities

In addition to all the standard LaTeX design elements, the TLP style includes the following feature:

- Extended commands for specifying a short version of the title and author(s) for the running headlines.

Once you have used this additional facility in your document, do not process it with a standard LaTeX style file.

2.1 Titles authors' names and affiliation

In the TLP style, the title of the article and the author's name (or authors' names) are used both at the beginning of the article for the main title and throughout the article as running headlines at the top of every page. The title is used on odd-numbered pages (rectos) and the author's name appears on even-numbered pages (versos). Although the main heading can run to several lines of text, the running head line must be a single line.

Moreover, the main heading can also incorporate new line commands (*e.g.* \\) but these are not acceptable in a running headline. To enable you to specify an alternative short title and author's name, the standard \lefttitle commands have been used to print the running headline. If more authors has to be used in \author command then each authors should be captured in separate \author command. \affiliation command is used to call the affiliation, if more affiliations has to be used in \affiliation command then each affiliations should be captured in separate \affiliation command. \email command should be used inside the affiliation as shown below.

```
\lefttitle{Cambridge Author}
\LaTeX\ Guidelines for~authors}
  \title{The full title which can be as long as necessary}
  \author{Author's name}
  \affiliation{the affiliation if necessary}
```

2.2 Abstract

The TLP style provides for an abstract which is produced by the following commands

```
  \begin{abstract}  ...  \end{abstract}
```

---

[1] To know more information about LaTeX and its packages, try https://ctan.org/?lang=en

2.3 Keyword

The TLP style provides for an Keyword which is produced by the following commands

```
\begin{keywords}  ...  \end{keywords}
```

2.4 Lists

The TLP style provides the three standard list environments.

- Bulleted lists, created using the `itemize` environment.
- Numbered lists, created using the `enumerate` environment.
- Labelled lists, created using the `description` environment.

2.5 Footnotes

The TLP journal style uses superior numbers for footnote references.[2]

3 Some guidelines for using standard facilities

The following notes may help you achieve the best effects with the TLP style file.

3.1 Sections

LaTeX provides five levels of section headings and they are all defined in the TLP style file:

- `\section`.
- `\subsection`.
- `\subsubsection`.
- `\paragraph`.
- `\subparagraph`.

Section numbers are given for sections, subsection and subsubsection headings.

3.2 Running headlines

As described above, the title of the article and the author's name (or authors' names) are used as running headlines at the top of every page. The title is used on odd-numbered pages (rectos) and the author's name appears on even-numbered pages (versos).

3.3 Tables

The `figure` and `table` environments are implemented as described in the LaTeX Manual to provide consecutively numbered floating inserts for illustrations and tables respectively. The standard inserts and their captions are formatted centred. Line breaks in captions can be inserted as required using \\.

---

[2] This shows how a footnote is typeset.

The TLP style file will cope with most positioning of your tables and you should not normally use the optional positional qualifiers on the `table` environment which would override these decisions. Normal journal style sets the table caption first, followed by a double rule, the table body and a double rule at the bottom. Single rules and spanner rules (`\cline`) can be used to separate headings from the columns. For example, Table 1 is produced using the following commands:

```
\begin{table}[h!]
 \centering
 \caption{Results of Overloading for 3 Experimental Setups}\label{sample-table}
 {\tablefont\begin{tabular}{@{\extracolsep{\fill}}lcrrrrr}
   \topline
    Program& Expt.&
     CPU\footnote{Seconds of elapsed time on an unloaded Sun 3/50.}&
     RelCPU\footnote{CPU Time relative to experiment (a).}&
     GC& Mem\footnote{Bytes of heap used over the duration of the program.}&
     RelMem\footnote{Memory usage relative to experient (a).}\midline
    8 Queens& (a)&   2 88&  1 00&     6&   1 7M&  1 00\\
    &         (b)&  32 51& 11 29&   193&  48 9M& 28 76\\
    &         (c)&   7 90&  2 74&    42&  11 3M&  6 65\\
    \multirow{4}{*}[3pt]{Primes}&   (a)&   4 89&  1 00&   19&   5 3M&  1 00\\
    &         (b)&  47 54&  9 72&   204&  54 5M& 10 28\\
    &         (c)&  10 08&  2 06&    47&  13 0M&  2 45\\
    Nfib&     (a)&  21 65&  1 00&   161&  40 4M&  1 00\\
    &         (b)& 221 65& 10 24& 1382& 349 0M&  8 64\\
    &         (c)&  21 30&  0 98&   161&  42 0M&  1 03\\
    KWIC&     (a)&   7 07&  1 00&    15&   6 3M&  1 00\\
    &         (b)&  34 55&  4 89&   109&  47 8M&  7 59\\
    &         (c)&  31 62&  4 47&    53&  45 0M&  7 14
   \botline
    \end{tabular}}
\end{table}
```

Notice the use of the macro to obtain the centered decimal points, inside the body of the table.

Table 1. *Results of Overloading for 3 Experimental Setups*

Program	Expt.	CPU[3]	RelCPU[4]	GC	Mem[5]	RelMem[6]
8 Queens	(a)	2 88	1 00	6	1 7M	1 00
	(b)	32 51	11 29	193	48 9M	28 76
	(c)	7 90	2 74	42	11 3M	6 65
	(a)	4 89	1 00	19	5 3M	1 00
Primes	(b)	47 54	9 72	204	54 5M	10 28
	(c)	10 08	2 06	47	13 0M	2 45
Nfib	(a)	21 65	1 00	161	40 4M	1 00
	(b)	221 65	10 24	1382	349 0M	8 64
	(c)	21 30	0 98	161	42 0M	1 03
KWIC	(a)	7 07	1 00	15	6 3M	1 00
	(b)	34 55	4 89	109	47 8M	7 59
	(c)	31 62	4 47	53	45 0M	7 14

The `tabular` environment should be used to produce ruled tables; it has been modified for the TLP style in the following ways:

Fig. 1. An example figure with space for artwork.

1. Additional vertical space is inserted above and below a horizontal rule (produced by `\hline`);
2. Tables are centred, and span the full width of the page; that is, they are similar to the tables that would be produced by `\begin{minipage} {\textwidth}`.

Because of this reformatting, vertical rules should not be used; furthermore, commands to redefine quantities such as `\arraystretch` should be omitted. If the old tabular facilities are needed, there is a new environment, `oldtabular`, which has none of the reformatting; it should be used in exactly the same way.

3.4 Illustrations (or figures)

The TLP style will cope with most positioning of your illustrations and you should not normally use the optional positional qualifiers on the `figure` environment which would override these decisions. Figure captions should be below the figure itself, therefore the `\caption` command should appear after the figure or space left for an illustration.

Figure 1 shows an example onw working with LaTeX code to load art files. `\includegraphics` commnad is to load art files `scale` option used in `\includegraphics` is to reduce the art. EPS format will be compiled using LaTeX. PNG, PDF and JPG format art files are loaded in the same command but the TeX file should be compiled using PDFLaTeX:

```
\begin{figure}
  \includegraphics[scale=.4]{sample.eps}
  \caption{An example figure with space for artwork.}
  \label{sample-figure}
\end{figure}
```

The vertical depth should correspond roughly to the artwork you will submit; it will be adjusted to fit the final artwork exactly.

3.5 Creating new theorem-like environments

You can create your own environments in LaTeX, and although you may already be familiar with `\newtheorem`, you will not have seen the other two commands explained below.

`\newtheorem` is a standard command used for creating new theorem-like environments, such as theorems, corollaries, lemmas, conjectures and propositions, with the body of the text (automatically) in italic.

4 Mathematics

The TLP class file will centre displayed mathematics, and will insert the correct space above and below if standard LaTeX commands are used; for example use \[... \] and *not* $$ ... $$. Do not leave blank lines above and below displayed equations unless a new paragraph is really intended.

amsmath.sty is common package to handle various type math equations. The amsmath descriptions are available in the document can be find in the web link https://ctan.org/pkg/amsmath?lang=en

4.1 Numbering of equations

The subequations and subeqnarray environments have been incorporated into the TLP class file (see Section 4.1.1 regarding the subequations environment). Using these two environments, you can number your equations (1a), (1b) etc. automatically. For example, you can typeset

$$a_1 \equiv (2\Omega M^2/x)^{\frac{1}{4}} y^{\frac{1}{2}} \tag{1a}$$

and

$$a_2 \equiv (x/2\Omega)^{\frac{1}{2}} k_y/M. \tag{1b}$$

by using the subequations environment as follows:

```
\begin{subequations}
\begin{equation}
  a_1 \equiv (2\Omega M^2/x)^{\textstyle\frac{1}{4}}
    y^{\textstyle\frac{1}{2}}\label{a1}
\end{equation}
and
\begin{equation}
  a_2 \equiv (x/2\Omega)^{\textstyle\frac{1}{2}}k_y/M.\label{a2}
\end{equation}
\end{subequations}
```

4.1.1 The `subequations` environment and the `AMSTEX` package

The amstex (and the amsmath) packages also define a subequations environment. The environment in TLP.cls is used by default, as the environments in the AMS packages don't produce the correct style of output.

Note that the subequations environment from the amstex package takes an argument – you should use an 'a' to give \alph style subequations. e.g.

```
\begin{subequations}{a}  ...  \end{subequations}
```

4.2 Bibliography

As with standard LaTeX, there are two ways of producing a bibliography; either by compiling a list of references by hand (using a thebibliography environment), or by using BibTeX with a suitable bibliographic database with the bibliography style provided

with the tlpguide.tex like `\bibliographystyle{tlplike}`. The tlp.bst will produce the bibliography which is similar to TLP style but not exactly. If any modification has to be made with tlp.bst can be adjusted during manuscript preparation but the updated bst file should be given with source files. However, contributors are encouraged to format their list of references style outlined in section 4.2.2 below.

4.2.1 References in the text

References in the text are given by author and date. Whichever method is used to produce the bibliography, the references in the text are done in the same way. Each bibliographical entry has a key, which is assigned by the author and used to refer to that entry in the text. There is one form of citation – `\cite{key}` – to produce the author and date. Thus, Salton *et al.* (1990) is produced by

`\cite{sal90}`.

`natbib.sty` is common package to handle various reference and its cross citations. The natbib descriptions are available in the document can be find in the web link `https://ctan.org/pkg/natbib?lang=en`

4.2.2 List of references

The following listing shows some references prepared in the style of the journal.

```
\begin{thebibliography}{}
\bibitem[\protect\citename{Akmajian and Lehrer, }1976]{akm76}
 Akmajian \& Lehrer A. 1976. NP-like quantifiers and the
 problem of determining the head of an NP. {\it Linguistic
 Analysis\/} {\it 11}, 1, 295--313.
\bibitem[\protect\citename{Huddleston, }1984]{hud84}
 Huddleston, Rodney. 1984. {\it Introduction to the Grammar of
 English}. Cambridge: Cambridge University Press.
\bibitem[\protect\citename{McCord, }1990]{mcc90}
 McCord, Michael C. 1990. Slot grammar: a system for simpler
 construction of practical natural language grammars. In R.
 Studer (ed.), {\it Natural Language and Logic: International
 Scientific Symposium}, pp.~118--45. Lecture Notes in Computer
 Science. Berlin: Springer-Verlag.
\bibitem[\protect\citename{Salton {\it et al.}, }1990]{sal90}
 Salton, Gerald, Zhao, Zhongnan \& Buckley, Chris. 1990.
 A simple syntactic approach for the generation of indexing
 phrases. Technical Report 90--1137, Department of Computer
 Science, Cornell University.
\end{thebibliography}
```

This list typesets as shown at the end of this guide. Each entry takes the form

```
\bibitem[\protect\citename{Author(s), }Date]{tag}
  Bibliography entry
```

where `Author(s)` should be the author names as they are cited in the text, `Date` is the date to be cited in the text, and `tag` is the tag that is to be used as an argument for the `\cite{}` command. `Bibliography entry` should be the material that is to appear in the bibliography, suitably formatted. This rather unwieldy scheme makes up for the lack of an author-date system in LaTeX.

4.3 Catchline and date commands

To be placed in the preamble; for example:

- `\jnlDoiYr{2021}`
- `\doival{10.1017/xxxxx}`
- `\jnlPage{1}{8}`

4.4 Editing citations (when the author has used the \cite command)

In the past when an automatic `\cite` command produced text in the output which
needed to be changed, the argument (in [ ]) from the bibliography entry was copied to
the location of the `\cite` command and then modified. The `\cite` command would then
be removed as part of this process.

In the near future, we will probably have to supply TeX output which will need to
contain 'PDF marks' for interactive browsing. Clearly by removing the automatic link to
the bibliographic entry (referenced by the `\cite`), we are making extra work for ourselves
later on.

To avoid this, the function of the `\cite` command's optional argument has been
changed. For example, the `\cite` command for the '`mcc90`' entry gives:

<div align="center">McCord (1990)</div>

but you want the following to appear in the text:

<div align="center">McCord (1990, see p. 119)</div>

you would then use:

<div align="center"><code>\cite[McCord (1990), see p.~119)]{mcc90}</code></div>

to obtain the desired result. Notice that you have to supply the round brackets as well
in the optional argument.

References

Akmajian & Lehrer A. 1976. NP-like quantifiers and the problem of determining the head of an
NP. *Linguistic Analysis 11*, 1, 295–313.

Huddleston, Rodney. 1984. *Introduction to the Grammar of English.* Cambridge: Cambridge
University Press.

McCord, Michael C. 1990. Slot grammar: a system for simpler construction of practical natural
language grammars. In R. Studer (ed.), *Natural Language and Logic: International Scientific
Symposium*, pp. 118–45. Lecture Notes in Computer Science. Berlin: Springer-Verlag.

Salton, Gerald, Zhao, Zhongnan & Buckley, Chris. 1990. A simple syntactic approach for the
generation of indexing phrases. Technical Report 90–1137, Department of Computer Science,
Cornell University.