

Formal Aspects of Computing: L^AT_EX 2_ε Guide for Authors

Mark Reed¹ and Christiane Notarmarco²

¹Electronic Products and Composition Group,
Cambridge University Press, Cambridge,

²Springer-Verlag London Limited, Godalming, Surrey, UK

Abstract. This guide is for authors who are preparing papers for the *Formal Aspects of Computing* journal using the L^AT_EX 2_ε document-preparation system and the Formal Aspects of Computing class file (`fac.cls`).

Keywords: L^AT_EX 2_ε; Class file; `fac.cls`; User guide

1. Introduction

In addition to the standard submission of hardcopy from authors, *Formal Aspects of Computing* now accepts machine-readable forms of papers in L^AT_EX 2_ε. The layout design for the *Formal Aspects of Computing* journal has been implemented as a L^AT_EX 2_ε class file, based on the `article` class as discussed in the L^AT_EX manual (2nd edition) [Lam94].

Commands which differ from the standard L^AT_EX interface, or which are provided in addition to the standard interface, are explained in this guide (which is *not* a substitute for the L^AT_EX manual itself).

Note that the final printed version of papers will use the Monotype Times typeface rather than the Computer Modern available to authors. For this reason line and page breaks will change and authors should not insert hard breaks in their text.

Authors planning to submit their papers in L^AT_EX 2_ε are advised to use `fac.cls` as early as possible in the creation of their files.

1.1. Introduction to L^AT_EX

L^AT_EX is constructed as a series of macros on top of the T_EX typesetting program. L^AT_EX adds to T_EX a collection of facilities which simplify typesetting for authors by allowing them to concentrate on the logical structure of the document rather than its visual layout. Careful use of the L^AT_EX mark-up philosophy results

Correspondence and offprint requests to: Christiane Notarmarco, Springer-Verlag London Limited, Sweetapple House, Catteshall Road, Godalming, Surrey GU7 3DJ, UK. e-mail: chris@svl.co.uk

in uniform layout rather than the *ad hoc* results of some word-processing systems. Authors are advised to let the defaults control font selection etc., rather than tinker themselves.

The \LaTeX system provides a consistent and comprehensive document preparation interface. Among other things, \LaTeX can automatically number list entries, equations, figures, tables and footnotes, as well as sections and subsections. Using this numbering system, bibliographic citations, page references and cross references to any other numbered entity (e.g. sections, equations, figures) are straightforward.

1.2. The FAC document class

The use of document classes allows a simple change of style (or style option) to transform the appearance of your document. The Springer-Verlag London Limited (SVL) FAC class preserves the standard \LaTeX interface such that any document which can be produced using the standard \LaTeX `article` class can also be produced with the FAC class. However, the measure (or width of text) is different from that for `ARTICLE`; therefore line breaks will change and it is possible that longer equations may need re-setting. Authors are urged to use `fac.cls` from the beginning of their document preparation, otherwise longer lines may require re-formatting at a later stage.

1.3. General style issues

Use of \LaTeX defaults will result in a pleasing uniformity of layout and font selection. Authors should resist the temptation to make *ad hoc* changes to these. Also avoid use of direct formatting unless really necessary. Papers will be edited as usual, and this process may be obstructed by the use of inserted line breaks, etc.

For general style issues, authors are referred to the ‘Preparation of manuscripts’ in the back cover of the journal. Authors who are interested in the details of style are referred to [But81] and [Chi69]. The language used in the journals is British English, and spelling should conform to this.

Use should be made of symbolic references (`\ref`) in order to protect against late changes of order, etc.

1.4. Submission of \LaTeX articles

Authors who intend to submit a \LaTeX article to FAC should obtain a copy of the journal’s class file `fac.cls`. This is available on request in the first instance from Springer-Verlag London Limited; remember to specify the type of media you require. Alternatively, you can download the files at the journal’s home page via <http://www.springeronline.com>¹

If you cannot obtain the proper journal class file, use the standard `article` class with the default ‘10pt’ option.

When submitting the final article, ensure that the following are included and are clearly labelled.

1. A hardcopy printout of the article.
2. The input file (exactly matching the hardcopy).
3. A copy of all user-defined macros.
4. If you have used \BIBTeX , the `.bib`, `.bbl` and `.bst` files that were used.
5. Any other files necessary to prepare the article for typesetting.

Submit this, if at all possible by e-mail, and the hardcopy by post to D. J. Cooke (see the inside-cover of the journal for correct e-mail and postal addresses). If you do not have access to e-mail, send the files on disc – PC or Macintosh $3\frac{1}{2}$ in. – along with the hard copy to D. J. Cooke.

¹ do a “SPRINGER SEARCH” for the journal name or a “SPRINGER SEARCH – BY ISBN/ISSN” and click on “Download LaTeX macro package (ZIP archive)”

2. Using the FAC class file

First, copy the file `fac.cls` into an appropriate subdirectory on your system. The FAC class is implemented as a complete document class, and *not* as an class option. In order to use the FAC class, replace `article` by `fac` in the `\documentclass` command at the beginning of your document: that is,

```
\documentclass{article}
```

is replaced by

```
\documentclass{fac}
```

Author-defined macros should be inserted before `\begin{document}`, or in a separate file and should be included with the submission. Authors must not change any of the macro definitions or parameters in `fac.cls`.

2.1. Document class options

In general, the following standard document class options should *not* be used with the FAC class file:

- `10pt`, `11pt` and `12pt` – unavailable;
- `twoside` is the default (`oneside` is disabled);
- `onecolumn` is the default (`twocolumn` is disabled);
- `titlepage` is not required and is disabled;
- `fleqn` and `leqno` should not be used (`fleqn` is already incorporated into the FAC class).

3. Additional facilities

In addition to all the standard L^AT_EX design elements, the FAC class includes the following features.

1. Additional commands for typesetting the title page. Extended commands for specifying a short version of the title and author(s) for the running headlines.
2. The `\correspond` and `\makecorrespond` commands for printing the name and address for further correspondence.
3. The `keywords` environment.
4. Extended `\newtheorem` command which can produce unnumbered environments.
5. Proof environment.
6. Control of enumerated lists.

Once you have used these additional facilities in your document, it can be processed only with `fac.cls`.

3.1. Titles and author's name

In the FAC 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, which should not be more than 48 characters and spaces, and an alternative short author's name, the standard `\title` and `\author` commands have been extended to take an optional argument to be used as the running headline. *e.g.*

```
\title[Formal Aspects of Computing: \LaTeXe\ Submissions]
      {Formal Aspects of Computing:\\}
```

```

\LaTeXe\ Guide for Authors}

\author[M. Reed and C. Notarmarco]
  {Mark Reed$^1$ and Christiane Notarmarco$^2$\\
   $^1$Electronic Products and Composition Group,\\
   Cambridge University Press,Cambridge,\\
   $^2$Springer-Verlag London Limited, Godalming, Surrey, UK}

```

You may wish to add a `\thanks` note, which produces a footnote to the title or author. In this case, the superscripts by the authors' names will be automatically generated.

3.2. Correspondence note

The footnote at the bottom of the first page is generated automatically, and begins with the words '*Correspondence and offprint requests to*'. The appropriate name and address must be given as an argument to the `\correspond` command in the preamble:

```

\correspond{Christiane Notarmarco, Springer-Verlag London
  Limited, Sweetapple House, Catteshall Road,
  Godalming, Surrey GU7 3DJ, UK.
  e-mail: chris@svl.co.uk}

```

This information is subsequently used by the `\makecorrespond` command. This may be placed anywhere after the `\begin{document}` but within the text of the first page. If you have other footnotes on the first page, the `\makecorrespond` command should follow them ensuring the output of this command is printed at the foot of the page.

If you omit the `\correspond` and `\makecorrespond` commands, the name in the correspondence note defaults to the short author's name.

3.3. Abstract and keywords

At the beginning of your article, the title should be generated in the usual way using the `\maketitle` command. Immediately following the title you should include an abstract followed by a list of keywords. For example, the titles for this guide were produced by the following source:

```

\maketitle

\begin{abstract}
This guide is for authors who are preparing papers for the
\emph{Formal Aspects of Computing} journal using the \LaTeXe\ %
document-preparation system and the Formal Aspects of Computing
class file (\texttt{fac.cls}).
\end{abstract}

\begin{keywords}
\LaTeXe; Class file; \verb"fac.cls"; User guide
\end{keywords}

```

3.4. Theorems

The FAC class provides an extended `\newtheorem` command which enables you to generate unnumbered environments. e.g. An unnumbered 'Theorem' can be typeset with the following:

```

\newtheorem{theorem}{Theorem}[section]
:

```

```

\begin{theorem}
This gives a normal numbered theorem.
\end{theorem}
\begin{theorem*}
This gives an unnumbered theorem.
\end{theorem*}

```

Which produces:

Theorem 3.1. This gives a normal numbered theorem.

Theorem. This gives an unnumbered theorem.

The preferred numbering scheme is for theorems to be numbered within sections, as 1.1, 1.2, 1.3, etc. In order to allow authors maximum flexibility in numbering and naming, *no* theorem-like environments are defined in `fac.cls`. Rather, you have to define each one yourself. Theorem-like environments include Theorem, Definition, Lemma, Corollary, and Proposition.

3.5. Proofs

A new environment exists for Proofs, e.g.

Proof. Noting that a state satisfying VS has no elements V_{xi} of V satisfying $0 < V_{xi} < 1$, it follows that the proof is just as the proof was for the corresponding case of *Bi-States*. \square

This was produced by the following code:

```

\begin{proof}
Noting that a state satisfying  $VS$  has no elements
 $V_{xi}$  of  $V$  satisfying  $0 < V_{xi} < 1$ , it follows
that the proof is just as the proof was for the
corresponding case of Bi-States.
\end{proof}

```

The end of proof marker \square is produced automatically. If you wish to omit this, use the `proof*` environment instead.

The proof environment can also take an optional argument which allows you to produce ‘special’ proofs, e.g.

Proof of Theorem 1.1. Noting that a state satisfying VS has no elements V_{xi} of V satisfying $0 < V_{xi} < 1$, it follows that the proof is just as the proof was for the corresponding case of *Bi-States*. \square

Which was produced like this:

```

\begin{proof}[Proof of Theorem~1.1.]
Noting that a state satisfying  $VS$  has no elements
 $V_{xi}$  of  $V$  satisfying  $0 < V_{xi} < 1$ , it follows
that the proof is just as the proof was for the
corresponding case of Bi-States.
\end{proof}

```

Notice that once the optional argument is used, you have to type all of the text which is to appear as the heading (including the full-stop).

3.6. Lists

The FAC class provides the three standard list environments:

- Numbered lists, created using the `enumerate` environment;
- Labelled lists, created using the `description` environment;

- Bulleted lists, created using the `itemize` environment.

The enumerated list numbers each list item with an arabic numeral. Alternative styles can be achieved by inserting a redefinition of the number labelling command after the `\begin{enumerate}`. For example, a list numbered with roman numerals inside parentheses can be produced by the following commands:

```
\begin{enumerate}
  \renewcommand\theenumi{\roman{enumi}}
  \renewcommand\labelenumi{(\theenumi)}
  \item first item
  :
\end{enumerate}
```

This produces the following list:

- (i) first item
- (ii) second item
- (iii) *etc...*

In the last example, the labels were pushed out into the margin because the standard list indentation is designed to be sufficient for arabic numerals rather than for the wider roman numerals. In order to enable different labels to be used more easily, the `enumerate` environment in the FAC class can be given an optional argument which (like the standard `thebibliography` environment) specifies the *widest label*. For example,

- (i) first item
- (ii) second item
- (iii) *etc...*

was produced by the following input:

```
\begin{enumerate}[(iii)]
  \renewcommand\theenumi{\roman{enumi}}
  \renewcommand\labelenumi{(\theenumi)}
  \item first item
  :
\end{enumerate}
```

4. Mathematics

The FAC class will set displayed mathematics with the correct indent provided you use the \LaTeX standard of open and closed square brackets as delimiters. The following equation

$$\sum_{i=1}^p \lambda_i = \text{trace}(\mathbf{S})$$

was typeset in the FAC style using the commands:

```
\[ \sum_{i=1}^p \lambda_i = \mathrm{trace}(\mathbf{S}) \]
```

Note the difference between the positioning of this equation and of the following centred equation

$$\alpha_{j+1} > \bar{\alpha} + ks_{\alpha}$$

which was (wrongly) typeset using double dollars as follows:

```
$$ \alpha_{j+1} > \bar{\alpha} + ks_{\alpha} $$
```

Also, do not leave blank lines above and below displayed equations unless a new paragraph is really intended.

5. Typeface

5.1. AMS fonts – especially if you do not have them

If you need symbols from the AMS font set but do not have them installed, you can ensure that they will be correctly typeset by taking the following steps. Set up user-defined macros that can be redefined by the typesetter to use the correct AMS macros. For example, the blackboard bold symbols, sometimes called shell or outline characters, are obtained with the AMS macro `\mathbb{. .}`. Instead, use a macro definition such as:

```
% replace font!
\newcommand\BbbE{\ensuremath{\mathsf{E}}} % Blackboard bold E
```

This substitutes a sans serif character where you want blackboard bold. You can typeset the input file and the typesetter is alerted to do the substitution.

The following example (which uses the `\providecommand` macro) will work without modification by the typesetter, because the `\providecommand` macro will not overwrite any existing `\mathbb` definition.

```
\providecommand\mathbb[1]{\ensuremath{\mathsf{#1}}}
...
\newcommand\BbbE{\mathbb{E}} % Blackboard bold E
```

Plain T_EX provides only `\leq` and `\geq` which typeset the Computer Modern symbols \leq and \geq , respectively. These will be redefined at typesetting to use the AMS symbols `\leqslant` and `\geqslant`, to give slanted symbols.

If you wish to use AMS fonts with L^AT_EX 2_ε you must be using at least version 2.0. Earlier versions are not supported.

5.2. Sans serif symbols

The `\textsf` and `\mathsf` commands change the typeface to sans serif, giving upright characters. Occasionally, other sans serif fonts are needed. You should use the following supplied macros to obtain these other fonts.

<code>\textsf{text}</code>	→	text	<code>\$_mathsf{math}\$</code>	→	math
<code>\textsf{i}{text}</code>	→	<i>text</i>	<code>\$_mathsfi{math}\$</code>	→	<i>math</i>
<code>\textsf{b}{text}</code>	→	text	<code>\$_mathsfb{math}\$</code>	→	math
<code>\textsf{bi}{text}</code>	→	<i>text</i>	<code>\$_mathsfbi{math}\$</code>	→	<i>math</i>

You can use them like this:

```
\newcommand\ssC{\mathsf{C}} % for sans serif C
\newcommand\sfsP{\mathsfi{P}} % for sans serif slanted P
\newcommand\sfbSX{\mathsfb{X}} % for sans serif bold slanted X
```

Note that the bold-slanted macros `\textsfbi` and `\mathsfbi` use the slanted sans serif font `cmssi` – because there is no bold-slanted math sans serif font available in Computer Modern! If you use the supplied sans-serif text and math commands the typesetter will be able to substitute the fonts automatically.

5.3. Bold math italic symbols

If you require bold math italic symbols/letters, L^AT_EX provides several ways of getting them. Firstly, L^AT_EX's `\boldmath` switch which can be used like this:

```
$ \mbox{\boldmath $P$} = \mathsf{J}\mathcal{E} T $
```

As you can see it takes quite a lot of typing to achieve a bold math italic P. Another problem is that you can't use `\boldmath` in math mode, thus the `\mbox` forces the resulting text into text style (which you may not want).

You can cut down on the typing by defining the following macro (which is in the preamble of this guide):

```
\providecommand\boldsymbol[1]{\mbox{\boldmath $#1$}}
```

Then for the above example you can define:

```
\newcommand\boldP{\boldsymbol{P}}
```

to achieve the same result. This still doesn't remove the default to text style problem.

If you have the `amsbsy` package on your system, then you can remove this limitation as well by placing a `\usepackage{amsbsy}` after the `\documentclass` line in your document, and then use `\boldsymbol` for bold math italic symbols (don't define `\boldsymbol` yourself when using `amsbsy`).

5.4. Script characters

Script characters should be typeset using $\LaTeX 2_{\epsilon}$'s `\mathcal` command. This produces Computer Modern symbols such as \mathcal{E} and \mathcal{F} in your hard copy but the typesetter will substitute the more florid script characters normally seen in the journal.

5.5. Skewing of accents

Accents such as hats, overbars and dots are normally centred over letters, but when these are italic or sloping greek the accent may need to be moved to the right so that it is centred over the top of the sloped letter. For example,

```
\newcommand\hatp{\skew3\hat{p}}
```

will produce \hat{p} .

6. User-defined macros

If you define your own macros, you must ensure that their names do not conflict with any existing macros in \LaTeX (or AMS \LaTeX if you are using this). You should also place them in the preamble of your input file, between the `\documentclass` line (but after any `\usepackage` commands) and before the `\begin{document}` command.

Apart from scanning the indexes of the relevant manuals, you can check whether a macro name is already used by using `\newcommand`, which will check for the existence of the macro you are trying to define. If the macro exists \LaTeX will respond with:

```
! LaTeX Error: Command ... already defined.
```

In this case you should choose another name, and try again.

Such macros must be in a place where they can easily be found and modified by the journal's editors or typesetter. They must be gathered together in the preamble of your input file, or in a separate `macros.tex` file with the command `\input{macros}` in the preamble. Macro definitions must not be scattered about your document where they are likely to be completely overlooked by the typesetter.

The same applies to font definitions that are based on Computer Modern fonts. These must be changed by the typesetter to use the journal's correct typeface. In this case, you should draw attention to these font definitions on the hard copy that you submit for publication and by placing a comment in your input file just before the relevant definitions, for example `% replace font!`

7. Some guidelines for using standard facilities

The following notes may help you achieve the best effects with the FAC class file.

Fig. 1. An example figure in which space has been left for the artwork.

7.1. Sections

L^AT_EX provides five levels of section headings and they are all defined in the FAC class file:

```
Heading A – \section{...}
Heading B – \subsection{...}
Heading C – \subsubsection{...}
Heading D – \paragraph{...}
Heading E – \subparagraph{...}
```

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

7.2. Illustrations (or figures)

The FAC class 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. See ‘Instructions for authors’ in the FAC journal for submission of artwork. Figure captions should be below the figure itself therefore the `\caption` command should appear after the figure or space left for an illustration. Any figure should be centred horizontally, this can be achieved using the `\centering` command, or `centerline` environment. For example, Figure 1 is produced using the following.

```
\begin{figure}
  \centering
  \vspace{5cm}
  \caption{An example figure in which space has been
           left for the artwork.}
  \label{sample-figure}
\end{figure}
```

7.3. Tables

The FAC class 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. Table captions should be at the top, therefore the `\caption` command should appear before the body of the table.

The `tabular` environment should be used to produce ruled tables; it has been modified for the FAC class in the following ways.

1. Rules may either be two-thirds or the full width of a page, depending on the width of the material in your table. `\hline` will produce a rule two-thirds the width and `\fullhline` will produce a rule the full width of a page;

Table 1. Index of Symbols

Symbol	Meaning
$\triangleright\blacktriangleright$	Forward closure
$\blacktriangleright\triangleright$	Backward closure
$\blacktriangleright\blacktriangleright$	Overlap closure
$\triangleright\triangleright$	Rule closure
\rightsquigarrow	Rule closure constructor
\triangle	Substitution sum
\cdot	Substitution composition
\sqcup	Substitution join
\perp	Coherence relation
$-^n$	Substitution operator (exponent)
$W_n(-, -)$	Substitution operator (Whale)
$T_n(-, -, -)$	Substitution operator (Turtle)

2. Additional vertical space is inserted on either side of a rule;
3. Vertical lines are not produced.

Commands to redefine quantities such as `\arraystretch` should be omitted. For example, Table 1 is produced using the following.

```

\begin{table}
  \caption{Index of Symbols}
  \begin{tabular}{@{}l@{}}
    \hline
    Symbol\hspace{1cm} & Meaning \\
    \hline
    \whbl & \hbox{Forward closure} \\
    \blwh & \hbox{Backward closure} \\
    \blbl & \hbox{Overlap closure} \\
    \whwh & \hbox{Rule closure} \\
    $\leadsto$ & \hbox{Rule closure constructor} \\
    $\bigtriangleup$ & \hbox{Substitution sum} \\
    $\cdot$ & \hbox{Substitution composition} \\
    $\sqcup$ & \hbox{Substitution join} \\
    $\perp$ & \hbox{Coherence relation} \\
    $-^n$ & \hbox{Substitution operator} \\
    & \hbox{(exponent)} \\
    $W_n(-, -)$ & \hbox{Substitution operator} \\
    & \hbox{(Whale)} \\
    $T_n(-, -, -)$ & \hbox{Substitution operator} \\
    & \hbox{(Turtle)} \\
    \hline
  \end{tabular}
  \label{symbols}
\end{table}

```

7.3.1. Continued captions

If a table or figure will not fit onto a single page and has to be broken into more than one part, the subsequent parts should have a ‘continued’ caption. To achieve this, use `\contcaption` instead of `\caption`, in the subsequent figure or table environments. The `\contcaption` uses exactly the same syntax as the normal `\caption` command, except it does not step the figure or table counter.

7.4. Appendices

You should use the standard L^AT_EX `\appendix` command to place any Appendices, normally, AFTER any acknowledgement and references. From that point on `\section` commands will produce appendixes, which are numbered A, B etc. Equations, figures and tables continue in the same numbering sequence as in the main text.

7.5. References

As with standard L^AT_EX, there are two ways of producing a list of references; either by compiling a list (using a `\thebibliography` environment), or by using Bib_TE_X with a suitable bibliographic database, using the standard Bib_TE_X ‘alpha’ style file.

7.5.1. References in the text

References to published literature should be quoted in text by an abbreviation in square brackets of name(s) (three letters) and date (two digits) – this format is consistent with the Bib_TE_X ‘alpha’ style.

Each entry has a key, which is assigned by the author and used to refer to that entry in the text (with `\cite`). e.g. `\cite{Lam:LaTeX}`.

A new form of the `\cite` command is also provided ‘`\nbcite`’, which works in the same way – except it doesn’t put any square brackets in the output, allowing more complex citations formats to be constructed.

7.5.2. The list of references

The following listing shows some references prepared in the style of the journal; the code produces the references at the end of this guide.

```
\begin{thebibliography}{Lam94}
  \bibitem[But81]{But:Copy} Butcher, J.:
    \emph{Copy-editing: The Cambridge Handbook.}
    Cambridge University Press, 1981.

  \bibitem[Chi69]{Chicago}
    \emph{The Chicago Manual of Style.}
    University of Chicago Press, Chicago 60637, USA, 1982.

  \bibitem[For84]{For:Program} Forgaard, R.:
    A Program for Generating and Analyzing Term Rewriting
    Systems, Master’s Thesis, MIT Lab. for Computer Science,
    1984.

  \bibitem[JLR82]{Jou:Recursive} Jouannaud, J. P., Lescanne, P.
    and Reinig, F.: Recursive Decomposition Ordering,
    \emph{Proc. Conf. on Formal Description of Programming
    Concepts II}, pp. 331--346, 1982.

  \bibitem[Lam94]{Lam:LaTeX} Lampion, L.:
    \emph{\LaTeX: A Document Preparation System (2nd edition).}
    Addison-Wesley, New York, 1994.

  \bibitem[Ped85]{Ped:Obtaining} Pederson, J.:
    Obtaining Complete Sets of Reproductions and Equations
    without Using Special Unification Algorithms. Unpublished
    manuscript, 1985.

  \bibitem[PeS81]{Pet:Complete} Peterson, G. E. and Stickel,
```

M. E.: Complete Sets of Reductions for Some Equational Theories. `\emph{J. ACM}`, `\textbf{28}`, 223--264 (1981).
`\end{thebibliography}`

Where more than one reference is cited having the author(s) and date, the letters a,b,c, ... should follow the date (*e.g.* [Smi88a], [Smi88b], *etc.*). References should be listed in `thebibliography` environment alphabetically by author(s)' name(s) and then by year if the same author has several papers.

Formatting for italic etc. should be avoided unless you are sure you understand the style of references; please concentrate on giving full and clear information.

References

- [But81] Butcher, J.: *Copy-editing: The Cambridge Handbook*. Cambridge University Press, 1981.
- [Chi69] *The Chicago Manual of Style*. University of Chicago Press, Chicago 60637, USA, 1982.
- [For84] Forgaard, R.: A Program for Generating and Analyzing Term Rewriting Systems, Master's Thesis, MIT Lab. for Computer Science, 1984.
- [JLR82] Jouannaud, J. P., Lescanne, P. and Reinig, F.: Recursive Decomposition Ordering, *Proc. Conf. on Formal Description of Programming Concepts II*, pp. 331--346, 1982.
- [Lam94] Lamport, L.: *L^AT_EX: A Document Preparation System (2nd edition)*. Addison-Wesley, New York, 1994.
- [Ped85] Pederson, J.: Obtaining Complete Sets of Reproductions and Equations without Using Special Unification Algorithms. Unpublished manuscript, 1985.
- [PeS81] Peterson, G. E. and Stickel, M. E.: Complete Sets of Reductions for Some Equational Theories. *J. ACM*, **28**, 223--264 (1981).

A. Special commands in `fac.cls`

The following is a summary of the new commands, optional arguments and environments which have been added to the standard L^AT_EX user-interface in creating the FAC class file.

New commands

<code>\contcaption</code>	produces a continued figure or table caption. e.g. ‘Figure 1— <i>cont.</i> ’ This command is used in the same way as the standard <code>\caption</code> command.
<code>\correspond</code>	sets the text used for the ‘Correspondence and offprint requests to’ footnote on the first page.
<code>\makecorrespond</code>	ensures the ‘Correspondence and offprint requests to’ footnote is typeset. This command is usually used just before <code>\maketitle</code> .
<code>\mathproofbox</code>	as <code>\proofbox</code> , except this version is intended for use in equations ending proof’s (it typesets the proof box using <code>\rlap</code> with 1em of extra space).
<code>\nbcite</code>	works in the same way as the normal <code>\cite</code> command except it doesn’t put in the ‘[]’s.
<code>\newtheorem</code>	has been modified to create both numbered and unnumbered ‘theorem’ environments.
<code>\proofbox</code>	typesets a proof box \square (this is normally put in automatically at the end of the <code>proof</code> environment). If you need to insert a <code>\proofbox</code> manually, you should add a <code>\quad</code> of space before it in the output.
<code>\removebrackets</code>	removes the ‘()’ brackets from the optional argument of environments created by the <code>\newtheorem</code> command. Should be placed just before the appropriate environment.

New environments

<code>proof</code>	to typeset mathematical proofs, the <code>*</code> -form omits the proof box.
<code>tabular</code>	has been modified to insert additional space above and below an <code>\hline</code> , the table caption and body is centred with rules full out across the text measure. The <code>tabular</code> environments old behaviour can be restored by using the <code>oldtabular</code> environment.

New optional arguments

<code>[<short author>]</code>	in the <code>\author</code> command: to define a left running headline that is different from the authors’ names as typeset at the article opening.
<code>[<short title>]</code>	in the <code>\title</code> command: to define a right running headline that is different from the article title.
<code>[<widest label>]</code>	in <code>\begin{enumerate}</code> : to ensure the correct alignment of numbered lists.