

TYPOGRAPHY IN COMPUTER SCIENCE USING T_EX

Arjen Rouvoet

DESIGNING TYPE is a beautiful job, combining the creative with the mechanical to make words appear in print. It is tempting for authors to forget the design aspect and only consider the mechanical, thus reducing *typography* to the act of *typesetting*. As an unfortunate consequence of this reduction, our conference proceedings are full of papers where the design of the type *takes away* from the content. Perhaps it is noticeably sloppy, or inconsistent, or downright wrong. My hypothesis is that authors tend to underestimate the effect of both good and bad typography. One way that I like to think about this in terms of the cognitive bias known as the halo effect. The human mind is constantly trying to form a coherent picture of the outside world. This effort is exaggerated¹, so that a positive impression of any entity in one area tends to color ones opinion in different areas. This is not a conscious consideration and it takes effort to overcome. Hence, concluding that a paper has beautiful, meaningful content, when it is presented as ugly, confusing type, is a lot to ask from a reader.²

Despite the importance of typography to ease knowledge transfer in science, we receive little training in the visual presentation of text. Thanks to Knuth, we have fantastic open source tools that simplify the mechanical aspects of the task^{3,4}, but as of today we have no tools that can automate the design of type: T_EX is not an automated typographer, it is a typesetting engine. T_EX sources are sequences of precise instructions for the engine, not a specification of meaning that can be turned into good typography. *You* are the typographer.

MY HOPE IS that this at least convinces you that the visual aspects of papers matter. Perhaps, however, you are skeptical that this concerns you, as an author. After all, do conferences not hand out templates that take care of the design? Indeed, we are lucky with a typesetting engine that supports the abstraction of many aspects of

¹ Kahneman 2012. “Thinking, Fast and Slow”

² Erik Spiekermann reminds the readers of Practical Topography of Paul Watzlawick’s first axiom of communication that says that it is not possible to not communicate. We may wish that a reader would ignore text appearance, but it is more realistic to think that they simply cannot.

³ Hofstadter 1979. “Gödel, Escher, Bach: an Eternal Golden Braid”

⁴ It is easy to underestimate just how much work T_EX saves us. Douglas R. Hofstadter gives us a glimpse of what it used to take to typeset a book in the early computer era.³ He describes how he painstakingly produced the printing galleys using the laborious computer-assisted, but still very manual process at the local printer. After working all summer getting the manuscript typeset he discovers that the acid used to develop the galleys was not adequately cleaned, slowly erasing the ink. The only solution was to repeatedly fly back to Stanford during in the weekends of his teaching term to redo the manual labor. Consider that next time that you curse T_EX for its (admittedly many) shortcomings.

type design. Yet, the overall appearance of papers is also determined by many considerations that are out of the scope of templates and the abstractions implemented using T_EX macros are not very strong. One needs to develop a good eye for judging and debugging the produce of the engine.

Both the fact that typography matters and the fact that it must concern authors is expressed succinctly by the definition of *good* typography by Matthew Butterick in his book *Practical Typography*:

“Good typography is measured by how well it reinforces the meaning of the text, not by some abstract scale of merit.”

Clearly we should be aware of all aspects within our control that can contribute or take away from the conveyed meaning of our writing. And clearly, if typography is about the meaning of text, it ought to involve the author.

THAT ALL BEING SAID, I have to be frank and let you know that I am by no means an expert in this area. Better documents than this exist that develop the topic from first principles. This document just aims to help you avoid the slow process of learning the basics by osmosis. It is a collection of notes to draw your attention to your tasks and responsibilities as an author when you set type using T_EX for a conference. The topics are chosen based on my personal struggles, and those that I recognize in conference proceedings. One can read, or scan, the notes linearly to get an idea of the scope, and subsequently use them for reference. It is a living document, and if you have suggestions for topics then I will gladly receive them.

It may be encouraging to know that a little bit of attention to the matter goes a long way. Despite the fact that I am not an expert, the small considerations that are summarized in this note produced a thesis that stood out to my audience and committee as a beautifully crafted document.⁵ Even the defense attendees that were not able to appreciate the content, appreciated how it looked. A good first impression invites reading and encourages the reader. This is the halo effect in action.

⁵ Rouvoet 2021. “Correcty by Construction Language Implementations”

IT IS INSPIRING to see examples of good typesetting. Consider the following (publicly accessible) documents:

- [The Microtype Manual](#)
- [Sample pages of “Trees, maps, and theorems”](#)

Setting a Page

► DISTURBED PAGES AND PARAGRAPHS: WIDOWS & ORPHANS

Paragraphs are indented—or typeset after a vertical skip—to mark their boundary. This enables your reader to see the structure of the text without doing any reading. There are several situations that can arise that disturb these visual clues.

A *widow* is the last line of a paragraph at the top of a page or column. An orphan is the first line of a paragraph at the bottom of a page. Such unfortunate placement disturbs the usual appearance of text layout on a page, and are better avoided. Depending on the situation one can cut a page short (`\clearpage`, `\newpage`), lengthen a page by some amount (`\enlargethispage{<length>}`), or do some massaging to make everything fit better.

Very short paragraphs (i.e., less than three lines) should be avoided for the same reason: the visual clues that are normally there to support the reader start looking erratic.

The same logic also applies to words in paragraphs: the last word of a paragraph on the front of a new line is also referred to as a widow, and one should seek to avoid it.

► BAD AND WORSE SPACE HACKING

One of the last thing that tends to happen before a paper deadline is “space hacking”: when an imposed page limit is not met, one goes around the paper and cramps figures and text in the least amount of space. Obviously, this tends to be at odds with the best practices of setting pages: optimizing content for the least amount of space taken is not optimizing for ease of reading. That being said, sometimes it just needs to be done. Please consider in those cases the following tips for bad (rather than worse) space hacking.

Consistency is the key to good page layout. Instead of sprinkling negative vertical space around the paper at random, consider adjusting them consistently using the predefined or custom lengths and macros. For example, instead of adding `\vskip{-.5cm}` after a bunch of figure captions to reduce the white space below it, consider one of the predefined lengths that determine the spacing globally:

```
\setlength{\floatsep}{5pt}
\setlength{\belowcaptionskip}{-10pt}
\setlength{\abovecaptionskip}{5pt}
```

► \VSPACE VS. \VSKIP

Do you know the difference between `\vspace{<length>}` and `\vskip <length>`? The difference can easily bite you if you use the wrong one. Compare this fragment:

There is a `\vspace{1em}` in the middle of this sentence.

with this one:

There is a `\vskip 1em`

in the middle of this sentence.

The point is that the former command inserts the space after the “current vertical box”, whereas the latter immediately ends the vertical box, inserts the vertical space, and starts a new vertical box. To understand exactly how this works, one needs to review how L^AT_EX interleaves its operation in horizontal and vertical mode.

My own experience is that I usually need vertical space *right here, right now*—e.g., because one needs more space around some element in the layout. In such cases, `\vskip` is more appropriate.

► A PALETTE OF SECTIONING TOOLS

Most article formats use indentation to mark a new paragraph. The indent indicates that some line of thought has come to an end, and that a new one is about to start. Within a section we have many more markers to delimit lines of thought and indicate pause. In increasing pause length:⁶

```
... end of paragraph. \smallskip
... end of paragraph. \medskip
... end of paragraph. \\
\paragraph{...}
\subsection{...}
```

It is a good idea to make a selection from this palette and stick to it. Do not mix `\paragraph{·}` and `\subsection{·}` arbitrarily. Pick a rule and stick to it.

⁶ In the Tufte-LaTeX common package there is `\newthought{·}`, which inserts a medium skip and sets the first words in small caps to visually distinguish the start of a new line of thought. Its function is similar to `\paragraph{·}`, but instead of a title, the first few words of the first sentence are visually distinguished.

Setting a Paragraph

► SPURIOUS INDENTATION

Paragraphs are a semantic notion: they span a certain line of thought. L^AT_EX automatically inserts paragraph demarcation based on whitespace in your source file. This, of course, requires some cooperation: avoid whitespace in the middle of a paragraph. Our papers tend to have paragraphs that span equations or code snippets (or enumerations, or theorems, or...). Around such display elements you may need to take extra care with paragraphing. Paragraphs can (and, semantically, sometimes should) continue after such a display. Either do not have whitespace in the source, comment out the whitespace using `%`, or use a `\noindent` after some whitespace to continue a paragraph:⁷

```
..., as follows:

\l f() + g() \l

\noindent
Here, the addition...
```

Do not be fooled by the idea that L^AT_EX should have an understanding of the structure of your document. Whitespace implies indentation. If you put whitespace before a theorem, the theorem will be indented. Whitespace before a minipage indents the whole minipage.

► MICROTYPE ADJUSTMENTS

Typesetting is a design discipline, not an exact science. If something does not look good, then it is hard to argue that it is good typesetting, even if it is pixel perfect. Conversely, you might “feel” that one text looks better, without understanding exactly why. There are some subtleties to typesetting paragraphs. For example, a margin optically appears better justified if punctuation protrudes into the margin. Such “micro type adjustments” are implemented in pdfT_EX and various other engines and can be enabled as follows:⁸

```
\usepackage{microtype}
```

The acmart document class requires it out of the box.

⁷ The last one always bugged me a little, because it is not semantic. This is an unnecessary concern. L^AT_EX sources are imperative typesetting scripts and most of the semantics that we bestow upon the source is illusory.

⁸ Thành Thế 2000. “Micro-typographic extensions to the T_EX typesetting system”

A justified paragraph without character protrusion can look ragged.

Nadat hij woorden en concepten heeft bekriftigd, spreekt het vanzelf dat Nietzsche doordendert: ook de verklaring is gebaseerd op een logica die schatplichtig is aan geloof (in woorden en concepten) en moet daarom worden afgeschaft. Hij verkiest de interpretatie, het onderzoek naar het ontstaan van waarden die ons omringen en waarvan wij de ontwikkeling ondervinden, boven het steeds maar weer proberen concepten te formuleren en sacraliseren, dat wil zeggen holle identiteiten die zogenaamd objectief zijn. We moeten niet langer het reële willen onthullen, niet langer proberen de sluier op te lichten die over ideeën ligt, waarvan wij denken dat ze onafhankelijk van ons bestaan (je herkent de kritiek op het platonisme). Het reële is er nog niet, het is niet neutraal en beschikbaar voor iedereen, maar het is onder onze ogen bezig te ontstaan op een unieke en individuele manier. Het vraagt erom opgepakt te worden, dus

► INDENT DISPLAYS IN PARAGRAPHS

Equations, code and other displays that are part of a paragraph but not boxed, should be indented or centered. Displays that appear flushed to the left margin of the body with surrounding whitespace appear as floats or by themselves. Indenting them makes it visually more apparent that they belong to the surrounding block.

Setting a Sentence

L^AT_EX tries very hard to let text have an optically uniform look. We want to avoid undoing this hard work by user errors.

► LINE-HEIGHT VIOLATIONS

Inline use of graphics, or mathematics, can sometimes violate the line-height restrictions, leading to ugly paragraphs.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi $\sum_{i=0}^{n^m18}$ architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit.

► OVERFULL AND UNDERFULL HBOXES

T_EX tries to optimize line breaking to get well justified paragraphs with uniform interword spacing. One measure that it considers is the *natural* width of lines, which is the width of a line before making typographic adjustments (interword spacing, font expansion, etc.) for justification. A big difference between the natural width and the `\textwidth` results in bad paragraphing. In such cases, T_EX reports underfull or overfull HBoxes, together with the amount.

Underfull HBoxes can also be caused by bad `\break` commands.

A very sparse line\break

A	very	sparse	line
---	------	--------	------

This is why, in case one really wants to break a line, an `\hfill` is in order:

```
A very sparse line\hfill\break
```

A very sparse line

Or, equivalently, use `\:`:

```
A very sparse line\:
```

A very sparse line

Both underfull and overfull HBoxes can also be caused by long unbreakable elements. A common source of unbreakable boxes is verbatim (e.g., due to `\lstinline`) text:

```
Let me end a sentence with a
    \verb|VeryLongCamelCaseIdentifierFactoryBean|
```

Let me end a sentence with a
 VeryLongCamelCaseIdentifierFactoryBean

This is another example of a typographic issue that has no mechanical solution. One simply has to be on the look-out for these occurrences, and rewrite sentences and paragraphs as needed to avoid awful breaks. Such is life.

► ITALICS CORRECTION

When one changes from an italic to an upright font, the space between the last italic character and the adjacent upright character will appear less than in the rest of the sentence. This inconsistency is corrected by the insertion of an *italics correction* that depends on the slanted character. The amount to be corrected is associated with each character in the font. It can be inserted using the command `\/`—for example:

```
{\it ROBOT}\KOP
```

*ROBOT*KOP

Compare this with:

```
{\it ROBOT}KOP
```

ROBOTKOP

Although in some cases the difference is not dramatic, it may be quite distracting in general, depending on the slant of the character. In the margin there are two more examples with a different font.

THE BEST WAY to deal with this in a typical scenario is to avoid the T_EX macros `\it` (and `\itshape`) for achieving italics and use the L^AT_EX macro `\textit{.}` instead. This macro will automatically insert italics correction. One still has to be aware in case other macros—such as color commands—interfere with automatic italics correction:

```
\textit{\textcolor{red}{ROBOT}}KOP
vs.
\textit{\textcolor{red}{ROBOT\}}KOP
```

ROBOTKOP vs. *ROBOTKOP*

► TITLES

Titles are either typeset in *title-case* or *sentence-case*.⁹ In the former style (used in this document) all words are capitalized except for short “minor” words. The cited APA style guide specifies it more precisely. In any case, pick a style and stick to it.

► QUOTATION MARKS

Quotes in L^AT_EX are directional!¹⁰ The following is wrong:^{11,12}

Not sure about “those” quotation marks, or “these” quotation marks.

```
Not sure about "those" quotation marks, or "'these'"
quotation marks.
```

Pretty sure about “them” quotes, which also come in ‘singles’.

```
Pretty sure about ``them'' quotes, which also come in `singles'.
```

Missing italics correction:

$i : \mathbb{N} \rightarrow$	$T)$
------------------------------	------

⁹ Association 2019. *APA Style: Title Case Capitalization*

¹⁰ Knuth’s T_EXbook treats quotes on the first technical page!

¹¹ Butterick 2010. “Practical Typography”

¹² The quotation marks depend on the language, but straight quotes are never correct. There are various packages available to help you out:

```
\usepackage{dirtytalk}
\usepackage{csquotes}
```


Language	LOC Spec	Tests	Succeed	Fail	Stuck
Java	1201	125	125	0	0
Scala	517	109	109	0	0
LMR	263	19	15	0	4
Total	1976	253	249	0	4

```

\begin{tabular}{l c c c c c}
\toprule
Language & LOC Spec & Tests & Succeed & Fail & Stuck \\
\midrule
Java & 1201 & 125 & 125&0&0 \\
Scala & 517 & 109 & 109&0&0 \\
LMR & 263 & 19 & 15&0&4 \\
\midrule
Total & 1976 & 253 & 249&0&4 \\
\bottomrule
\end{tabular}

```

The headers and footers are visually clearly distinguished. The table feels spacious because there are no rules between the rows in the body. Extra padding is used around the rules that mark the header and footer to avoid cramping.

Math

► TEXT IN MATHMODE

A ubiquitous T_EX sin is inclusion of text in mathmode without marking it as text. This yields rendered text with very bad kerning, because mathmode renders it not as text, but as juxtaposed individual characters:

$$myfunction(x, y) + awful(z)$$

```
\[ myfunction(x, y) + awful(z) \]
```

Any text in a math environments needs to be explicitly switched to text mode. One can switch to a text mode with an appropriate style using one of the `\text{,it,sf,bf}` macros. My preference is towards using `\textsf` for typesetting function names, distinguishing them clearly from variables, which are typeset in italic:

$$myfunction(x, y) + not-so-awful(z)$$

```
\[ \textsf{myfunction}(x, y) + \textsf{not-so-awful}(z) \]
```

The macro `\text{.}` is well suited for comments and descriptions in math environments:

```
\usepackage{amsmath}
```

$$f(x, y) = \sum_{i=x}^y i \quad \text{My definition.}$$

Mathematical functions and other symbolic names are not code (usually), and should not be typeset in monospace fonts. They are also not a normal word in a sentence, and hence should not be typeset using `\text{.}`.

► OPERATORS

► ALIGNMENT

► SYMBOL MACROS

Symbolic mathematics can be hard to edit because equations become very macro-dense. To avoid this, one can define Unicode characters:

```
% preamble
\usepackage{newunicodechar}
\newunicodechar{\emptyset}{\ensuremath{\emptyset}}
\newunicodechar{\vdash}{\ensuremath{\vdash}}
\newunicodechar{\alpha}{\ensuremath{\alpha}}
\newunicodechar{\beta}{\ensuremath{\beta}}
\newunicodechar{\rightarrow}{\ensuremath{\rightarrow}}

% body
\[ \emptyset \vdash t : \alpha \rightarrow \beta \]
% instead of:
\[ \emptyset \vdash t : \alpha \rightarrow \beta \]
```

This helps to avoid typos, as the source corresponds more closely to the typeset result. Do make sure that your coauthors also know how to enter Unicode symbols.

► INFERENCE RULES

There are several packages for typesetting inference rules and natural deduction proofs.

My recommendation is the `mathpartir` package by Didier Rémy. Inference rules are set like this (when used in conjunction with `\newunicodechar`):

```
\usepackage{mathpartir}
\usepackage{bussproofs}
\usepackage{semantic}
```

```
\begin{mathpar}
  \inferrule[T-Fun]
  {\Gamma \vdash f : a \rightarrow b} \ \text{and} \ \Gamma \vdash e : a
  {\Gamma \vdash f e : b}
  \and
  \inferrule[T-Var]
  {\Gamma(x) = a}
  {\Gamma \vdash x : a}
\end{mathpar}
```

$$\frac{\text{T-FUN} \quad \Gamma \vdash f : a \quad \Gamma \vdash e : a}{\Gamma \vdash fe : b}$$

$$\frac{\text{T-VAR} \quad \Gamma(x) = a}{\Gamma \vdash x : a}$$

Fonts

There is a lot to be said about fonts,¹³ but I am not in a position to give advice here. Luckily, fonts are generally determined by the document class prescribed by the conference. I want to draw your attention to a handful of pitfalls:

¹³ Butterick 2010. “Practical Typography” (Font Recommendations)

► MONOSPACE FONT FOR CODE

The default monospace font is not always great. Packages are available with different monospace fonts for use in L^AT_EX.

```
\usepackage{inconsolata}
\usepackage[scaled]{beramono}
```

► FONT IN FIGURES

When you create figures containing text outside of L^AT_EX, make sure you select a matching font.

► MIXING FONTS

Fonts have a feel. Missing symbols that are replaced by symbols from a different font, or symbols that are handcrafted can easily feel out of place:

Figure 1: Jarring ampersands in action in mathmode.



$$!A \stackrel{\text{def}}{=} \bigotimes_{n \in \mathbb{N}} A^{\otimes n} / \sim_n \quad H_A(X) \stackrel{\text{def}}{=} \perp \& A \& (X \wp X)$$

Here we have an example from a (generally well typeset) paper that used an ampersand symbol that is either from a different font, or handcrafted. Not only does it not match the feel of the other symbols (e.g., by having a different weight, and no serifs), it also drops below the baseline of the sentence.

► TYPEWRITER CONSIDERED TOXIC

Typewriter fonts have no place outside code snippets.

► SWITCHING FONTS IN TITLES

Titles are visually distinguished by the font. Beware of macros that switch the font in a title as they may disrupt the overall appearance too much.

Bad Macros

Technical Notes

► EMBEDDED FONTS

To make sure that your pdf looks the same everywhere, fonts should be embedded in the document. This should be the case for most, if not all, packaged fonts. It may occur, however, that you use a (unicode) character that is not included in any of the packaged fonts, in which case things may go awry. You can check that your document only uses embedded fonts using any decent pdf reader. In Evince, for example, it can be found under *Properties > Fonts*.

Other Resources

Why should we care about typography and typesetting? In the end it is the content that matters, is it not? The introduction of this document attempted to give some motivation, but others have presented much better arguments than I:

- [Why does typography matter.](#)

References

Hofstadter, Douglas R (1979). “Gödel, Escher, Bach: an Eternal Golden Braid”. Twentieth-anniversary. Basic Books, Inc., New York.

Thàn Thê, Hàn (2000). “Micro-typographic extensions to the TEX typesetting system”. PhD thesis. Masaryk University, CZ. URL: <https://tex.loria.fr/moteurs/han-the-thanh-thesis-lin.pdf>.

Butterick, Matthew (2010). “Practical Typography”. URL: <https://practicaltypography.com/>.

Kahneman, Daniel (2012). “Thinking, Fast and Slow”. Penguin Books.

Association, American Psychological (2019). *APA Style: Title Case Capitalization*. URL: <https://apastyle.apa.org/style-grammar-guidelines/capitalization/title-case> (visited on 09/23/2021).

Rouvoet, Arjen (2021). “Correcty by Construction Language Implementations”. PhD thesis. Delft University of Technology, NL. URL: <https://ajrouvoet.github.io/files/thesis.pdf>.