George Grätzer

Practical LaTeX

EXTRA MATERIALS
extras.springer.com

Springer

Practical LaTeX

George Grätzer

Practical LaTeX

Springer

George Grätzer
Toronto, ON, Canada

Additional material to this book can be downloaded from http://extras.springer.com

ISBN 978-3-319-06424-6 ISBN 978-3-319-06425-3 (eBook)
DOI 10.1007/978-3-319-06425-3
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014942524

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my family
and especially the little ones,
Emma (9),
Kate (7),
Jay (2)

Short Contents

Contents

Introduction

To learn LaTeX, you have to read some really heavy books, such as my big book on LaTeX[1] (I will refer to it as MiL4), all 600 plus pages of it. These books are so big because they cover all of LaTeX, from everyday use to complex documents, such as books, and the esoteric.

This book is a practical introduction to LaTeX. It covers only what is most used in everyday documents. If you want to learn how to typeset fine-tuning, read MiL4. Chances are slim that you would need this in your work.

We start with a lightning fast introduction to LaTeX.

Chapter 1 is *Mission Impossible*, introducing LaTeX documents and presentations in about 30 pages. After reading this chapter, you should be able to type your own documents and make your own presentations.

The other chapters delve deeper into the topics started in the first. Chapter 2 deals with typing text and Chapter 3 with text environments, such as lists and theorems. Chapter 4 deals with typing formulas and Chapter 5 with displayed formulas. The structure of a LaTeX document is discussed in greater detail in Chapter 6. LaTeX is so efficient to use because we can customize it to our needs, as discussed in Chapter 7.

We further develop our skills in making presentations in Chapter 8 and drawing illustrations in Chapter 9.

The text and math symbol tables are collected in Appendices A and B. We provide you with a pdf file `SymbolTables.pdf` (see Section 1.1.2), so you can have these tables handy on your computer's desktop.

Finally, in Appendix C, we show you how to use LaTeX on an iPad.

We achieve such a slim book by focusing on the contemporary and the practical. We don't write about legacy commands (such as `\bf`, use `\textbf`), environments (such as `eqnarray`, use `align`), and document classes (such as `article`, use `amsart`). There is no discussion of how to write a complex document such as a book, the fonts you can use, and of the various tools we have for long documents.

[1] *More Math into LaTeX*, 4th edition. Springer-Verlag, New York, 2007. ISBN-13: 978-0-387-32289-6

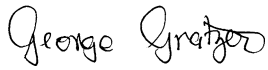
These topics would deserve separate *Practical* books. For further reading, see the file `FurtherReading.pdf` in the `samples` folder; see Section 1.1.2.

You will judge this book by how well it serves you. I selected the topics based on my experience writing articles and books in LaTeX and about LaTeX, and running an international math journal. I believe that the topics you need to type average size LaTeX documents are covered. If you have any thoughts about what else should be included, please let me know.

Acknowledgement

I received valuable advice for this book from William Adams, Jacques Crémer, Michael Doob, Alan Litchfield, Raul Martinez, Craig Platt, and Herbert Schulz.

Barbara Beeton is always there when I need her.



E-mail:
gratzer@me.com
Home page:
http://server.maths.umanitoba.ca/homepages/gratzer/

CHAPTER

1

Mission Impossible

It happens to most of us. We live a happy life without LaTeX and then, all of a sudden, we have to do something urgent that requires it.

If you are a student, maybe your professor turned to you and said "I need the solutions to these exercises typed up and distributed to the class by tomorrow" and the solutions are chock-full of formulas, difficult to do in Word.

Or you are a researcher whose documents have always been typed up by a secretary. You have to attend a conference and give a presentation. Your secretary is gone due to a budget cut ...

In my case, it was a letter (this was before e-mail) from the American Mathematical Society, in which they informed me that my paper, written in Word, was accepted for publication. The AMS will publish the paper in nine months. However a LaTeX version would be published in three months!

> The mission, should you choose to accept it, is to get started really fast in LaTeX. Our goal is to produce in LaTeX the little article printed on the next page.

Relax, this chapter will not self-destruct in five seconds.

G. Grätzer, *Practical LaTeX*, DOI 10.1007/978-3-319-06425-3_1,

A TECHNICAL RESULT FOR CONGRUENCES OF FINITE LATTICES

G. GRÄTZER

Abstract. We present a technical result for congruences on finite lattices.

1. Introduction

In some recent research, G. Czédli and I, see [1] and [2], spent quite an effort in proving that some equivalence relations on a planar semimodular lattice are congruences. The number of cases we had to consider was dramatically cut by the following result.

Theorem 1. *Let L be a finite lattice. Let δ be an equivalence relation on L with intervals as equivalence classes. Then δ is a congruence relation iff the following condition and its dual hold:*

(C_+) *If x is covered by $y, z \in L$ and $x \equiv y \pmod{\delta}$, then $z \equiv y + z \pmod{\delta}$.*

2. The proof

We prove the join-substitution property: if $x \leq y$ and $x \equiv y \pmod{\delta}$, then

$$x + z \equiv y + z \pmod{\delta}. \tag{1}$$

Let $U = [x, y + z]$. We induct on $\operatorname{length} U$, the length of U.

Let $I = [y_1, y + z]$ and $J = [z_1, y + z]$. Then $\operatorname{length} I$ and $\operatorname{length} J < \operatorname{length} U$. Hence, the induction hypothesis applies to I and $\delta \restriction I$, and we obtain that $w \equiv y + w \pmod{\delta}$. By the transitivity of δ, we conclude that

$$z_1 \equiv y + w \pmod{\delta}. \tag{2}$$

Therefore, applying the induction hypothesis to J and $\delta \restriction J$, we conclude (1).

References

[1] G. Czédli, *Patch extensions and trajectory colorings of slim rectangular lattices.* Algebra Universalis **88** (2013), 255–280.

[2] G. Grätzer, *Congruences of fork extensions of lattices.* Acta Sci. Math. (Szeged), **57** (2014), 417–434.

Department of Mathematics, University of Manitoba, Winnipeg, MB R3T 2N2, Canada
E-mail address, G. Grätzer: gratzer@me.com
URL, G. Grätzer: http://tinyurl.com/gratzerhomepage

Date: March 21, 2014.
2010 *Mathematics Subject Classification.* Primary: 06B10.
Key words and phrases. finite lattice, congruence.

1.1 Getting started

1.1.1 Your LaTeX

Are you sitting in front of your computer, your LaTeX implementation up and running? If you use a UNIX computer, you surely are. If you are in front of a PC (with the Windows operating system) or a Mac, point your Internet browser at `tug.org`. Choose to download MikTeX for a PC and MacTeX for a Mac. Follow the easy instructions (and be patient, these are big downloads) and you are done.

Even better, find a friend who can help.

On a PC, `work\test` refers to the subfolder `test` of the folder `work`. On a UNIX computer and on a Mac, `work/test` designates this subfolder. To avoid having to write every subfolder twice, we use `work/test`, with apologies to our PC readers.

1.1.2 Sample files

We work with a few sample documents. Download them from `CTAN.org`, search for `Practical LaTeX`, or go to the Springer page for this book, and click on the link:
http://extras.springer.com/2014/978-3-319-06424-6+

I suggest you create a folder, `samples`, on your computer to store the downloaded sample files, and another folder called `work`, where you will keep your working files. Copy the documents from the `samples` to the `work` folder as needed. *In this book, the* `samples` *and* `work` *folders refer to the folders you have created.*

One of the sample files is `sample.sty`. Make sure it is in the `work` folder when you typeset a sample document.

1.1.3 Editing cycle

Watch a friend type a document in LaTeX and learn the basic steps.

1. *A text editor is used to create a LaTeX source file*. A source file might look like this:

```
\documentclass{amsart}
\begin{document}
Then $\delta$ is a congruence relation. I can type formulas!
\end{document}
```

   Note that the source file is different from a typical word processor file. All characters are displayed in the same font and size.

2. *Your friend "typesets" the source file* (*tells the application to produce a typeset version*) *and views the result on the monitor*:

   Then δ is a congruence relation. I can type formulas!

3. *The editing cycle continues.* Your friend goes back and forth between the source file and the typeset version, making changes and observing the results of these changes.

4. *The file is viewed/printed.* View the typeset version as a pdf file, print it if necessary, to create a paper version.

If LaTeX finds a mistake when typesetting the source file, it records this in the *log file*. The *log window* (some call it *console*) displays a shorter version.

Various LaTeX implementations have different names for the source file, the text editor, the typeset file, the typeset window, the `log` file, and the `log` window. Become familiar with these names, so you can follow along with our discussions.

1.1.4 Typing the source file

A source file is made up of *text, formulas*, and *instructions* (*commands*) *to* LaTeX.

For instance, consider the following variant of the first sentence of this paragraph:

```
A source file is made up of text, formulas (e.g.,
$\sqrt{5}$), and \emph{instructions to} \LaTeX.
```

This typesets as

A source file is made up of text, formulas (e.g., $\sqrt{5}$), and *instructions to* LaTeX.

In this sentence, the first part

```
A source file is made up of text, formulas (e.g.,
```

is text. Then

```
$\sqrt{5}$
```

is a formula

```
), and
```

is text again. Finally,

```
\emph{instructions to} \LaTeX.
```

are instructions. The instruction `\emph` is a *command with an argument,* while the instruction `\LaTeX` is a *command without an argument.* Commands, as a rule, start with a backslash (\ ) and tell LaTeX to do something special. In this case, the command `\emph` emphasizes its *argument* (the text between the braces). Another kind of instruction to LaTeX is called an *environment.* For instance, the commands

```
\begin{center}   \end{center}
```

enclose a `center` environment; the *contents* (the text typed between these two commands) are centered when typeset.

In practice, text, formulas, and instructions (commands) are mixed. For example,

```
My first integral: $\int \zeta^{2}(x) \, dx$.
```

is a mixture of all three; it typesets as

My first integral: $\int \zeta^2(x)\,dx$.

Creating a document in LaTeX requires that we type in the source file. So we start with the keyboard, proceed to type a short note, and learn some simple rules for typing text in LaTeX.

1.2 The keyboard

The following keys are used to type the source file:

```
a-z   A-Z   0-9
+ = * / ( ) [ ]
```

You can also use the following punctuation marks:

```
,  ;  .  ?  !  :  ‘  ’  -
```

and the space bar, the Tab key, and the Return (or Enter) key.

Finally, there are thirteen special keys that are mostly used in LaTeX commands:

```
#  $  %  &  ~  _  ^  \  {  }  @  "  |
```

If you need to have these characters typeset in your document, there are commands to produce them. For instance, $ is typed as `\$`, the underscore, _, is typed as `\_`, and % is typed as `\%`. Only @ requires no special command, type `@` to print @; see Section A.3.3.

There are also commands to produce composite characters, such as accented characters, for example ä, which is typed as `\"{a}`. LaTeX prohibits the use of other keys on your keyboard unless you have special support for it. See the text accent table in Section A.2. If you want to use accented characters in your source file, then you must use the `inputenc` package.

Practical Tip 1. The text accent table looks formidable. Don't even dream of memorizing it. You will need very few. When you need a text accent, look it up. I know only one: `\"a` (LOL). If you have a name with accented characters, figure out once how to type it, and then any time you need it you can just copy and paste (chances are that the name is in your list of references).

1.3 Your first text notes

We start our discussion on how to type a note in LaTeX with a simple example. Suppose you want to use LaTeX to produce the following:

> It is of some concern to me that the terminology used in multi-section math courses is not uniform.
>
> In several sections of the course on matrix theory, the term "hamiltonian-reduced" is used. I, personally, would rather call these "hyper-simple". I invite others to comment on this problem.

To produce this typeset document, create a new file in your `work` folder with the name `textnote1.tex`. Type the following, including the spacing and linebreaks shown, but not the line numbers:

```
1  % Sample file: textnote1.tex
2  \documentclass{sample}
3
4  \begin{document}
5  It is of some concern to me   that
6  the terminology used in  multi-section
7   math courses is not uniform.
8
9  In several sections of the course on
10 matrix theory, the  term
11  ‘‘hamiltonian-reduced’’ is used.
12  I, personally, would rather call these
13 ‘‘hyper-simple’’. I invite others
14  to comment on this  problem.
15 \end{document}
```

Alternatively, copy the `textnote1.tex` file from the `samples` folder; see page 3.

The first line of `textnote1.tex` starts with `%`. Such lines are called *comments* and are ignored by LaTeX. Commenting is very useful. For example, if you want to add some notes to your source file and you do not want those notes to appear in the typeset version of your document, begin those lines with a `%`. You can also comment out part of a line:

```
simply put, we believe % actually, it's not so simple
```

Everything on the line after the `%` character is ignored by LaTeX.

Line 2 specifies the *document class*, `sample` (the special class we provided for the sample documents), which controls how the document is formatted.

The text of the note is typed within the `document` environment, that is, between `\begin{document}` and `\end{document}`.

Now typeset `textnote1.tex`. You should get the typeset document as shown. As you can see from this example, LaTeX is different from a word processor. It disregards the way you input and position the text, and follows only the formatting instructions given by the document class and the markup commands. LaTeX notices when you put a blank space in the text, but it ignores *how many blank spaces* have been typed. LaTeX does not distinguish between a blank space (hitting the space bar), a tab (hitting the Tab key), and a *single* carriage return (hitting Return once). However, hitting Return twice gives a blank line; *one or more* blank lines mark the end of a paragraph.

LaTeX, by default, fully justifies text by placing a flexible amount of space between words—the *interword space*—and a somewhat larger space between sentences—the *intersentence space*. If you have to force an interword space, you can use the `\␣` command (in LaTeX books, we use the symbol ␣ to mean a blank space). The `~` (tilde) command also forces an interword space, but with a difference: it keeps the words on the same line. This command produces a *tie* or *nonbreakable space*.