LaTeX 2_ε for Technion Graduate Students*

JOSEPH (YOSSI) GIL† MY OTHER SELF‡

Department of Computer Science
Technion—Israel Institute of Technology
Technion City, Haifa 32000, Israel

YoGi | yogi@cs.technion.ac.il

November 24, 2002

Abstract

Every paper starts with an abstract which should be short and written in a very special style. The abstract is used mostly by people who do not wish to read the paper, and are looking for short description of what's in it. The abstract is written last. In the first papers your write, it is usually better to let your advisor write it for you.

1 Introduction

In general, every paper has an introduction. The introduction puts the problem at hand in context. Why is it interesting, what were the previous attacks on it, and how a solution to it might be used.

I noticed that I spend (some would say waste) so much of my time in teaching the same basic writing skills to almost every graduate student who sought the dubious pleasures of my advisorship. Topic such as using LaTeX2e, basic typography, tabulating data, drawing clear figures, making precise mathematical definitions, and general scientific writing style, etc. are recurring so frequently, that I decided I must do something about it. Finding myself rewriting almost everything that a student writes, is frustrating both for me and for the student. Call me perfectionist, but that's how I am ...

Some people I work with may even end up with a bitter taste seeing everything they did torn apart, thinking that the fault lies with them. It is so difficult to accept the ancient proverb on the weight of words, and its consequences—that the amount of labor it takes to produce any written word is so much greater than typing it. Perhaps the best metaphor here is typing with the nose ...

In the interest of reducing my tutoring effort, I offered a seminar to a selected bunch of graduate students, mostly by invitation. This low quality document is the product of the first run of this seminar. It was not surprising to me to find out that most of the difficulties students incur have nothing to do with English, but rather with turning our rambling thoughts into a precise, focused and terse written document.

*This is a live document. Check often in `http://www.cs.technion.ac.il/ yogi/howto.tex` to find the latest version

†Work done in part during a visit to the IBM T.J. Watson Research Center

‡Contact author

The topics discussed included:

1. TEX, LATEX, and LATEX2e. Tools of the trade, and tricks of the gurus.
2. Elementary typography, fonts and simple checklists
3. BIBTEX, bibliography and references to bibliography.
4. The Elements of style by White and Strunk, the Elements of Grammar
5. The structure of sentences, paragraphs, and sections.
6. Paragraph structure
7. Paper outline: Title, abstract, keywords, introduction, conclusions, discussion, and future research, acknowledgements
8. Informative tables and figures
9. Formulae, definitions and theorems
10. Reviewing papers

Each participant will be asked to attend at least 75% of all meetings, prepare a summary of at least one topic. Weekly assignments included:

- Make a complete bibliographic entry of these five articles.
- Rewrite this sentence in 10 different ways.
- Translate this document into LATEX 2_ε.
- Point out all typographic errors in this paragraph.
- Tear apart this paragraph.
- Rewrite this paragraph.
- Identify the points made in each paragraph of a given section
- Criticize the writing of another course participant

Lectures were interactive, and I will frequently make use of a monitor to demonstrate the process of thinking and editing a scientific document.

Outline The last paragraph in the introduction should serve as a mini table of contents. In Sec. ??[1], we do so and so. Section blah does that, while Section tralaha does this. Finally, section what-ever-its-number is gives the conclusions.

Section:so-an

[1]Note that in giving reference to a section, correct technical writing dictates that we must capitalize the word "Section"

2 Basics

2.1 How to run LaTeX 2ε

Running LaTeX 2ε To compile an input LaTeX 2ε file

```
latex2e howto.tex
```

If this does not work, make sure that

```
/usr/local/lib/texmf/bin/sparc-solaris2.4
```

is in your path. Try to put it as early in the path list as possible. This should be done by including the following line in your `.login` file.

```
set path = (/usr/local/lib/texmf/bin/sparc-solaris2.4 $path)
```

Dealing with LaTeX 2ε Errors Do not loose heart. LaTeX 2ε is easy and most bugs can be tracked down very quickly. At first it is easier and faster to ask someone for help. Shortly thereafter, you will find that you can deal with errors yourself.

Multiple Runs If LaTeX 2ε produces warnings that some labels are undefined, you must run it again for the following reason. In the first run, LaTeX 2ε produces a file named `howto.aux` which includes definitions of the labels (as produced by the `\label` command). In the second run, LaTeX 2ε reads this file before processing the main input. This makes it possible to give forward references to labels. In the second run, the `.aux` file is reproduced, and this reproduction should be identical to the first. If they are different, LaTeX 2ε complains that "labels may have changed". You should rerun it to make it happy. The chances that you would require a fourth run are infintisimal.

It is interesting to examine the LaTeX 2ε definitions in the `.aux` file:

```
more howto.aux
```

In extreme situations, a buggy `.aux` file is produced, and LaTeX 2ε is not able to run at all as a result. In this case, you should remove the file `.aux` file.

```
rm howto.aux
```

and start over again.

Viewing The product of a LaTeX 2ε run is a `.dvi` (device independent) file. To view a `.dvi` file

```
xdvi howto.dvi
```

The default extension is `.dvi`, so for short, you could write:

```
xdvi howto
```

Usually, `xdvi` is not set up properly, to deal with A4 paper. Therefore, the correct command to run is:

```
xdvi -paper a4 howto
```

Alternatively, you can set up an alias or fix your `.Xdefaults` file, but I am not going to tell you how you should do any of these simple UNIX tasks ....

One problem you might have in viewing the file is that you did not set the `DISPLAY` environment variable correctly. This should be done by a command of the sort

```
setenv DISPLAY machine-name:0
```

For example, if the name of your workstation is `sg237s8` you should write:

```
setenv DISPLAY sg237s8:0
```

Converting to PostScript To create `howto.ps` from `howto.dvi`.

```
dvips howto.dvi
```

Again, the default is to search for a `.dvi` file, so the `.dvi` extension needs not be typed.

Viewing a PostScript file To view textttthowto.ps on your display, use

```
ghostview howto.ps
```

Again, the default is to search for a `.dvi` file, so the `.dvi` extension needs not be typed.

Printing To print a `.dvi` file on a PostScript printer, use the `-P` flag:

```
dvips -Pb617d2 howto.dvi
```

This will send the material to the `b617d2` printer, which is a black and white printer, located at room 617 in the 6^{th} floor, and is a duplex printer. Another way of doing this is:

```
dvips -Pb617d2 howto
```

for short. If you already generated a PostScript file then you can send it to the printer.

```
lpr -Pb617d2 howto.ps
```

Note that this time the extension has to be typed in. `lpr` does not have a default extension.

Bibliography To run BIBTEX:

```
bibtex howto
```

Note that BIBTEX requires multiple runs, sometimes as many as three.

2.2 Check list

Before showing your output to any other person, and especially before presenting it to an impatient advisor, make sure that your document is as perfect as it can be. Use the following check list:

1. Use LATEX 2_ε, rather than LATEX.
2. Spell check your document using `ispell`. Make sure that `ispell` produces a “clean” run! If it doesn’t, repeatedly add the extra words to your personal dictionary.

3. Use `lacheck` UNIX utility to discover some LaTeX 2$_\varepsilon$ input errors. Make sure that `lacheck` produces a "clean" run!

4. Can you find the typos in the following text?

   > In view of this painful possibility, I will not (as I might) appeal indignantly to my other writings as as a proof that I am incapable of such a deed: I will not (as I might) point to the strong moral purpose of this poem itself, to the arithmetical principles so cautiously inculcated in it, or to its its noble teachings in Natural History–I will take the more prosaic course of simply explaining how it it happened. [2]

   Hard to find? Check again! The words "as" and "its" are duplicated, a mistake usually happening with a line break between the two occurrences. Use my `dup` to eliminate duplicate words in sequence. The complete path to the `dup` utility is:

   ```
   /home/yogi/Bin/dup howto.tex
   ```

   As an extra bonus, `dup` also checks that all words which follow a period are capitalized, and it tries to apply a number of other little checks to your document.

5. Capitalization is also an important issue. Every sentence should start with an uppercase letter. The important words in titles are usually capitalized. This includes the first and last words and all other words except articles and prepositions. Proper nouns such us names of people or places should also be capitalized. Initials should be entirely in uppercase, e.g., FBI.

6. The acronyms "*i.e.*" and "*e.g.*" must be followed by a comma. Not including a comma is not only a grammatical error. It will also make TeX think that the last period in "*e.g.*" is a sentence end, adding funny extra space after it.l `dup` does try to this sort of errors.

7. One problem that `dup` cannot always detect is the correct use of articles. The indefinite article *a* should be used when followed by an initial consonant sound, including a pronounced *h*. For example: a computer, a division, a house.

   When the word starts with a vowel or a very weakly pronounced *h* (especially in British English), the article should be *an*. For example: an arch, an honor, an historian. The *an* article should be also used before a single letter (or initials) whose name starts with a vowel. For example, one should write *An* MS-DOS typical fault, *an* 8-bit field, etc.

8. Make sure that the input file uses indentation in such a way that reveals its logical structure. Use my `format` to apply smart indentation in the format suggested below.

   ```
   /home/yogi/Bin/format < howto.tex > howto.new.tex
   ```

   Manually check `howto.new.tex` because `format` is quite dumb.

[2] Quoted from "The Hunting of the Snark" by Lewis Carroll

3 Plain Text

In general, you should not worry yourself details of format of such things as space between paragraphs, line spacing, indentation and the such. It is a common mistake of novices to use low level commands such as `\\`, (forcing a line break), `\hspace` (forcing a horizontal space), `\vspace` (forcing a vertical space), etc. in their input to give it some desirable special appearance. You should resist the temptation to do so! Instead, you must search for the appropriate high-level, logical commands, which express the structure of your text, rather than any desirable visual appearance. For example, you should break a line explicitly, which is a physical operation. The correct *logical operation* is the start of a new paragraph, which is simply denoted by an empty line in the input.

The default output is set by the document class, and in almost all cases is what you want. In fairness, it should be said the defaults are what you should have wanted, if you spent enough time in learning all the bits and ends of typography. For example, in most document classes, paragraphs are indented in, with the exception of a first paragraph in a section or a subsection. Also, the vertical spacing between paragraphs is usually slightly larger in between lines.

Use indentation in your input file wisely, so that the logical structure would be apparent from the text layout. My suggestion is to start every sentence in a new line. If the sentence spans more than 72 characters, start the next line with an indentation. This gives you a visual clue on the number of sentences in each paragraph, which is usually 4–6.

Most of your LaTeX input should be plain text, with scattered mathematical formulae. Just type your text in, paying little or no attention to details such as line breaks, spacing between words, etc.

In typing your text, take note that there are some characters do have special meaning. The alphanumerical characters: A. . . ,Z, a,. . . , z, 0, . . . , 9 and most punctuation marks including the comma (,), the period (,), the semicolon (;), the exclamation mark (!), the question mark (?), etc. be typed directly into LaTeX 2ε. Table `Table:format:characters` 3.1 enumerates what we call *formatting* characters, used for placing ligatures and other symbols insider your text. (There are also some *special* characters which do not directly produce output; these will be the subject of the next section.)

Name	Appearance	Semantics	Escape
Period	`.`	Sentence end	`.\`
Greater than	`>`	Spanish exclamation mark (¡)	`$>$`
Less than	`<`	Spanish question mark (¿)	`$<$`
Open single quote	`` ` ``	Quoting	N/A
Close single quote	`'`	Quoting	N/A
Closing double quote	`"`	Quoting	N/A
Hyphen	`-`	Hyphenation	`{-}`
En dash	`--`	Range	`{-}{-}`
Em dash	`---`, `|`	Punctuation mark	`-{-}-`, `\textbar`

Table 3.1: Formatting characters `Table:`

3.1 Punctuation

The period (.) character, when followed by a space, denotes the end of the sentence. TeX places a little extra space between sentences[3]. There are however cases in which a period followed by a space is used to

[3] unless `\frenchspacing` is used.

denote an abbreviation, as in abbreviations of Latin words, including "vs.' *versus*, "etc." (etcetera), "et al." (et alii). The space following the period has to be escaped in these cases. For example, a figure caption "Runtime vs. input size" should be typed

```
Runtime vs.\ input size
```

Similarly, do not write the abbreviation "etc." simply as `etc. `, but rather as `etc.\ `, unless it comes at the end of the sentence.

The "less than" (<) and "greater than" symbol do not appear in normal text. A weird exception worth noting is that electronic mail programs tend to add a > symbol in front of the word "From" if it is the first word in a line.

Take note that displayed mathematical equations are part of the sentence and should usually be followed by a punctuation mark. Here is a simple example:

> Using Euler's equation,
>
> $$e^{i\pi} + 1 = 0,$$
>
> and the basic trigonometric equality, we obtain that
>
> $$\pi = \Leftrightarrow i \ln(\Leftrightarrow \sin^2 x \Leftrightarrow \cos^2 x).$$

Never write three dots `...`. Instead, use the `\ldots` macro. In mathematics, remember that a comma should precede and follow …, as in

$$\mathbf{x} = \langle x_1, \ldots, x_n \rangle.$$

Similarly, an operator should precede and follow `\cdots`:

$$1 + \frac{1}{2} + \cdots + \frac{1}{n} = H_n \approx \ln n + \gamma.$$

3.2 Quoting

LaTeX 2_ε distinguished between opening and closing quotes. Do not "quote" like this. Here is an example of correct "quoting". Thus, opening quotes should be typed in LaTeX as `` ` ` ``, and closing ones as `''`. Quotes should be used for quotations not as a substitute for an appropriate term, or loose presentation. `Teletype` font is the best way of marking verbatim text.

3.3 Dashes

The great variety in dashes and hyphens take is a major source of confusion even for experienced authors. One common mistake is to place a space before or after a hyphen or a dash. The different variations are:

- *Inter-word dash, entered as a single `-` character.* This dash is used to connect two words together, in phrases such as multiple-inheritance, and object-oriented entered as `multiple-inheritance` and `object-oriented`. Note that there is *no space* before or after an inter-word dash. Also note that many such phrases have evolved to be a single word, including familiar terms such as "database", "subtyping", and even "email"[4] Do not let yourself be distracted by spell checkers which are not aware of this language evolution. Do load a specialized dictionary of jargon and terminology.

  It is common and effective practice to factor out the second part of a conjugated phrase, as in "single- and multiple-inheritance", entered as

[4]The reader may want to check the strong position that D. E. Knuth takes in his web page against "e-mail".

```
single- and multiple-inheritance
```

- *Range specifier, or an* n-dash. The n-dash is used in phrases chapters I–III, pages 11–87.

  ```
  The n-dash is used in phrases chapters I--III, pages 11--87.
  ```

  Again, no space should be put before or after the dash.

- *The* m-dash *punctuation mark.* The m-dash, entered as three is used to separate parts of a sentence—in cases where a weaker than a semicolon but stronger than a comma separator is desired. It is entered as three consecutive minus characters, or as a vertical bar (|), with no spaces around it. Also, the official name of our institution is written

  The Technion—Israel Institute of Technology

  and typed in as

  ```
  The Technion---Israel Institute of Technology
  ```

  or

  ```
  The Technion|Israel Institute of Technology
  ```

Two other related variations are:

- *Minus sign in mathematical formulae.* A unary or a binary minus in mathematical formulae is entered simple as a minus sign. Spaces do not matter in mathematical formulae.

- *Discretionary hyphen, entered as* `\-` TEX does a pretty good job in breaking lines at word boundary and in hyphenating words when this not possible. In the extremely rare cases in which the correct hyphenation is different than what TEX algorithm thinks, you may insert `\-` anywhere in a word to designate an optional hyphenation point.

  The only case in which I found this to be necessary is in writing long URL addresses, as in `http://-www.cs.technion.ac.il/~yogi/howto.tex`, which is the web address of this document. However, now I prefer to write this address as footnote[5].

In summary, use dashes correctly. A double dash in the input is used to denote a range. A triple dash is used as punctuation—breaking a sentence into fragments. Note that there is no space before and after a triple dash. A single dash is used as a hyphen.

4 Special Characters

Several characters of the ASCII alphabet have a special meaning for LaTeX 2_ε. Table 4.1 enumerates all these characters.

`Table:command:characte`

[5] `http://www.cs.technion.ac.il/~yogi/howto.tex`

Name	Appearance	Semantics	Escape
Percent	%	Comment	\%
Backslash	\	Command prefix	\textbackslash
Open curly bracket	{	Grouping	\{
Close curly bracket	}	Grouping	\}
Hash sign	#	Argument	\#
Tilde	~	Non-breakable space	\textasciitilde
Dollar symbol	$	Inline formula	\$
Underscore	_	Subscript	_
Caret	^	Superscript	\verb+^+
Ampersand	&	Alignment	\&

Table 4.1: Special characters

Table:

4.1 Comments

As can be seen from the table, the percent symbol (%) is used for commenting. All text that follows a % until the end of line is not processed by TEX. It is easy to forget that the commenting carries a little further beyond the end of the line, and includes all white space characters which follow it:

```
Spacing away space%notice the use of spaces
    ships will become a major problem of the space age.
```

Spacing away spaceships will become a major problem in the space age.

4.2 Commands and Environments

Perhaps the most common special character is the backslash, which is used as the prefix of all LATEX *commands*. A command begins with a backslash, which can be followed by either a single, non alphabetical character (as in the \$ command), or any number of alphabetical characters. The use of backslash in ordinary text is rare. If you really need one, use \textbackslash. Take special note that \\ does not produce a "\", but rather breaks the current line. As mentioned above, it is a bad habit to use \\.

Curly brackets are primarily used for denoting *parameters* to commands. Thus for example, you may want to *emphasize* a piece of text by typing:

```
Against stupidity, the \emph{very gods themselves} contend in vain!
```

Against stupidity, the *very gods themselves* contend in vain!

Optional parameters are denoted by square brackets. The following command specifies that the default font size of the document is 12pt

```
\documentclass[12pt]{article}
```

Another use of curly brackets is for delimiting the scope of commands and definitions.

```
She grew {\small smaller and smaller} as she ate \ldots
```

She grew smaller and smaller as she ate ...

Curly brackets do not delimit scope when used for embracing parameters. If this is what you need, use another, nested, pair of curly brackets.

Parameters to most commands cannot span multiple paragraphs. *Environments* are used for applying commands to large text bodies. An environment is written as a \begin{x} \end{x} pair, where x is the environment's name.

4.2.1 The `document` Environment

A LaTeX 2ε document consists a *preamble* and a *body*. The preamble consists of global definitions. Some commands such as `\usepackage` can only be placed in the preamble. It is an error to include text intended for output in the preamble.

All document text is written inside the `document` environment:

```
% This is the preamble
\documentclass[12pt]{article}
\title{The Principle of Reciprocation}
\author{Mother Nature}
\usepackage{times}
\begin{document}
\maketitle
No good deed goes unpunished!
\end{document}
Anything written after \end{document} is ignored.
```

4.2.2 The `abstract` Environment

The `article` document class has an `abstract` environment:

```
\begin{abstract}
The abstract gives a short account of the
context and the main contributions of the
paper. It should be informative enough
so that experts can quickly determine
the relevance of the results to their
research.

Even though most abstracts span one
paragraph, abstracts of two and even
more paragraphs do exist. Some publishers
limit the abstract to 150 or so words.
\end{abstract}
```

Abstract

The abstract gives a short account of the context and the main contributions of the paper. It should be informative enough so that experts can quickly determine the relevance of the paper to their research.

Even though most abstracts span one paragraph, abstracts of two and even more paragraphs do exist. Some publishers limit the abstract to 150 or so words.

4.2.3 Environments as Scope Delimiters

All environments delimit the scope of any commands and definitions made in them.

```
\begin{quote}
\it Love is all you need!
\end{quote}
But love does not pay the rent.
```

Love is all you need!

But love does not pay the rent.

In the rare cases that this effect is not desired, one can invoke an environment using a command-like syntax:

```
\quote
\it Love is all you need!
\endquote
But love does not pay the rent.
```

Love is all you need!

But love does not pay the rent.

4.2.4 Making your own commands

New users are usually advised against defining their own commands, since in most cases, there are ready made packages to implement specialized needs. Do not trouble your little head in making complex macros.

Most of these were invented and can be found in the LaTeX 2_ε companion. However, it is important to define common symbols and terms as macros, to save typing, but much more importantly, to allow changes of notation.

A zoologist interested in spiders may therefore write:

```
\usepackage{xspace}
\newcommand{\spr}{spider\xspace}
\newcommand{\sps}{spiders\xspace}
\newcommand{\Spr}{Spider\xspace}
\newcommand{\Sprs}{Spiders\xspace}
...
\begin{document}
...
Another \spr came along, and then there were two \sps
    around, which made the situation pretty scary.
\end{document}
```

Which will result in

> Another spider came along, and then there were two spiders around, which made the situation pretty scary.

The `\xspace` command, defined in the `xspace` package, is designed to combat the annoying habit of commands of swallowing all subsequent spaces. Normally, when a command expands to a word, one should follow it by `{}` as in `\LaTeXe{}`. The `\xspace` command inserts a space, except when the subsequent character is a punctuation mark.

Here is a definition of a common symbol in mathematical writing:

```
\newcommand{\Dtu}{\Delta u}
```

This can be used later as:

```
\[\Dtu = f\] is a Poisson equation. For $f \equiv 0$
we get a Laplace equation: \[\Dtu = 0.\].
```

Which would produce the following output:

$$\Delta u = f$$

is a Poisson equation. For $f \equiv 0$ we get a Laplace equation:

$$\Delta u = 0.$$

If you try to use `\newcommand` to redefine an existing command, be it yours, or a LaTeX 2_ε builitin, you will get an error message. In the rare cases you wish to override existing commands, use `\renewcommand`.

The above definition of `\Dtu`will only work within math mode. If you wish to use it outside math mode, you will need to use `\ensuremath`, as follows:

```
\newcommand{\Dtu}{\ensuremath{\Delta u}}.
```

Experienced writers may define macros as shorthand, as in:

```
\newcommand{\beq}{\begin{equation}}
\newcommand{\eeq}{\end{equation}}
```

which would make `\beq ... \eeq` equivalent to

```
\begin{equation} ... \end{equation}.
```

It is a good idea to write all the definitions in a separate file and include it using the `\input` comand. Another alternative is to collect all commands in the preamble. This makes it easy to locate a definition and to quickly change a notation.

Here is a useful macro with a single argument:

```
\newcommand{\keyword}[1]{\texttt{\textbf{#1}}}
```

Take note how that the hash character (#) is used for denoting arguments to commands. With the above definition, one may write:

```
``\keyword{static}
    is a heavily overloaded keyword of C++''.
```

"**`static`** is a heavily overloaded keyword of C++".

4.3 Non-breakable Space

There are three important cases in which `~`(tilde) to denote non-breakable space.

- *Before inlined mathematical formulae.*

  ```
  Let~$i = 1, \ldots, n$, then it is
      not so difficult to prove that~$i \le n$.
  ```

  Let $i = 1, \ldots, n$, then it is not so difficult to prove that $i \leq n$.

- *Before citations.* The `\cite` command is used for citations. So, you should always write

  ```
  As Gil and Lorenz~\cite{Gil:Lorenz:96} ...
  ```

- Before references, i.e., `\ref`. To give a label to a figure, table, section, etc, write:

  ```
  \section{\LaTeXe{} Do's and Don'ts}
  \label{do-sec}
  ```

  The label command gives a name to the value of the last counter which LaTeX 2_ε used.

  Then, to give reference to that section, you can write:

  ```
  In Section~\ref{do-sec} we do blah blah.
  ```

  Note that in referencing a section by number, the word "section" must be capitalized. The same applies to theorems, lemmata, figures, tables, etc.

If you want to prevent a line break within a certain word, use the `\mbox` command. The following is a definition of a `\CC` command whose output is a non-breakable "C++".

```
\newcommand{\CC}{{C++}\xspace}
```

5 Organizing Your Article

5.1 Sectioning

Articles are broken invariably partitioned into sections, which are sometimes partitioned into subsections. The `\section` and the `\subsection` commands, which take their title as an argument, are used in making these divisions. Fig. 5.1 gives a general schematic structure of a LaTeX paper, and the use of `\section` to break it into sections.

Figure:structure

Also seen in the figure is `\paragraph` command, which is yet another partitioning command, which in most document classes is formatted as a single paragraph with no numbering. The `\appendix` command does not produce any output. However, sectioning commands that show up after it are numbered as appendices (A, B, ... in the defaults `article` document class).

Students have the tendency of over-sectioning, so perhaps I should not tell you that there is also a `\subsubsection` command, standing in the hierarchy between `\subsection` and `\paragraph`. In fact, I even regret telling you about the `\subsection` command.[6] Try to let your text flow smoothly; only break it apart if necessary. As a rule of thumb, a subsection should have four or even paragraph, and be introduced only if approved by your advisor.

Longer documents, such as dissertations, have chapters (`\chapter`).

Continue by defining the terms. What do you mean by 2-dim?, what is upward planar.

5.2 The Components of a Paper

Figure:structure

Fig. 5.1 shows all the mandatory textual elements of a scientific article.

Title The *Title* should be informative, and not too fancy or cute. Too special a title might create an unprofessional impression.

Authors In this part, name the authors and their affiliations. Affiliations are necessary since they give both information to the readers, and credit to the fonders. Ordering of authors is significant: usually it is alphabetical. Other ordering emphasizes the author listed first.

Abstract The *abstract* summarizes the paper in the domain's specific language. Since the paper is judged first according to the abstract, and can be rejected based on it alone, it should list the most innovative and dramatic points of the paper.

The abstract is searched by search-engines, for which purpose it can include a section of *keywords*, which determine the subject of the paper. The abstract may include *specifications* - indices in information-hierarchy trees managed by journals.

Introduction The *introduction* is intended for the reader who is familiar with the problem's domain. In this section the writers convince the reader that the problem is worth-while and innovative, and the solution exceeds existing solutions. Failure in proving these points shall lead to rejection of the paper. When no former solutions exist, the introduction should prove the problem to be of scientific interest.

Outline The *outline* appears as the last paragraph of the Introduction, or as a separate paragraph. It describes the different sections in the paper, and shows how they achieve the goals set in the introduction.

The outline is a list of sentences of the form *in Section 2 we show this*, and *Section 3 includes that*. Use the word *we*, even for a single author. To avoid boredom, use both passive and active forms,

[6] Naturally, I won't even mention the `\subparagraph` command.

```
\documentclass[12pt]{article}
\title{The Ultimate Answer}
\author{Douglas Hitch and Adam Hiking}
\usepackage{times}
\maketitle
\begin{document}
\begin{abstract} The problem of life ... \end{abstract}
\section{Introduction}
Ever through the years, mankind was bothered ...
\paragraph{Outline}
\section{Life} ...
\section{Universe} ...
\section{Everything} ...
\section{Conclusions and Open Problems}
We have shown that the answer is 42.
All that remains is to find the right question.
\paragraph{Acknowledgments}
We thank Agatha Christie for comments made on an
    earlier version of this document.
\bibliography{practice,book,crossref,latex}
\bibliographystyle{abbrv}
\appendix
\section{Proof of Goldbach's Conjecture}
\end{document}
```

Figure 5.1: The schematic structure of a LaTeX 2_ε article

Figure

along with various different verbs such as *describe*, *present*, *show*, *outline*, *review*, *discuss*, *prove*, and also *generalize* and *conclude*.

Write *Section* instead of *section*, as well as *Figure*, *Table*, *Theorem*, *Lemma* and, rarely, *Equation*. Use the word *Section* even when referring to a subsection, such as *Section 3.2*.

A special case in which we use *theorem* (lowercase *t*) when naming a specific theorem, such as *Murphy Brown's last theorem*.

Concluding Section Various names are used for the final section of a paper: *Conclusions*, *Concluding Remarks*, *Discussion*, *Open Problems*, *Further Research* and any combinations thereof.

Conclusions often summarize the paper. Including this part in a paper is done mostly for historical reasons: when papers used to describe scientific experiments, the Conclusions section named the conclusions of the experiments' results.

Nowadays, having the papers describe different subjects, the Conclusions Section should shed a new light on what has been stated in the introduction, based on the thorough understanding of the subject

The *Further Research* Section lists the problems remaining open—the most difficult and interesting ones.

Acknowledgements

Bibliography

Use last names only. A full name gives special honor to authors.

5.3 Lists

LaTeX 2_ε provides several important mechanisms for an hierarchical organization of your text. There are sevral ways to organize The first way is itemization:

- Itemizing:
- this is an item
- another item

Itemization is created in commands as the following:

```
\begin{itemize}
\item{Itemizing:}
\item{this is an item}
\item{another item}
\end{itemize}
```

The second way is enumeration:

1. Enumeration:
2. this is an item
3. another item

Where the commands for creating this enumeration are:

```
\begin{enumerate}
\item{Enumeration:}
\item{this is an item}
\item{another item}
\end{enumerate}
```

The third way is using descriptions:

Description:

this is an item

another item

Where the commands for creating this enumeration are:

```
\begin{description}
\item{Description:}
\item{this is an item}
\item{another item}
\end{description}
```

Too deep an indentation is not recommended, usually use not more than two levels.

6 Installing New Packages

kages

Since so many other researchers use LaTeX 2ε extensively, it is very likely that most of your customization needs have already been implemented by others. All you need to do is search for the solution, which almost always shows up as a *package*, sometimes called a *style file*. There are packages for inserting POSTSCRIPT graphics, enriched layout of table, enhanced footnotes, typesetting music, commutative diagrams, barcodes, Tamil characters, and many many more. Your LaTeX 2ε installation comes with a rather minimal collection of packages. If what you need cannot be found among these, just search the CTAN archives (`http://www.ctan.org`). A good search tool is available as

```
http://ring.asahi-net.or.jp/pub/text/CTAN/help/Catalogue/brief.html
```

Sometimes you may even want to define and use your own packages, or just install packages downloaded from the net or borrowed from a friend. A case in point is my `labels` package, which you can use by writing

```
\usepackage{labels}
```

The tricky bit is that you need to help LaTeX 2ε find the package, which is not standard. The file `labels.sty` is found in

/home/yogi/TeX

You could use a symbolic link to that directory, but an easier way to do that would be to set your `TEXINPUTS` environment variable.

```
setenv TEXINPUTS .:/home/yogi/TeX:
```

This tells TEX to search for macro files and packages first in the current directory, then in my macro directory `/home/yogi/TeX` and then in the standard system location of macro files. The final colon (`:`) is crucial. It tells LATEX 2_ε to continue the search for macro files in the standard system path. Without it, LATEX 2_ε will not be able to process even the most simple inputs. You may also want to create your directory with your own collected packages. Do no repeat the mistake of so many others in placing the packages in the same directory as the inputs.

If you are using the WinEdt/MikTeX combination on personal Windows system, you will need to do the following: Place the new package in the `localtexmf` directory, which is usually

```
C:\MikTeX\localtexmf\tex\latex
```

or

```
C:\Program Files\MikTeX\localtexmf\tex\latex
```

If the new package likes to reside in its own directory, just place it there.

Then you would need to tell MikTeX that a new package is installed in its path. To do so, you would need to run the following command from the DOS prompt.

```
initexmf --update-fndb
```

`initexmf` is usually found in one of the following two directories:

```
C:\MikTeX\texmf\miktex\bin+

C:\Program Files\MikTeX\texmf\miktex\bin+
```

You would need to be in that directory to run this command.

6.1 The `labels` Package

abels

Using the `labels` package, there is a much simpler way of defining labels and making references. A section is declared by a command like:

```
\section[do]{\LaTeXe{} Do's and Don'ts}
```

where `do` serves as the section label. In general, the capitalized version of the command takes a label as a second argument. Thus, there are also `\Subsection`, `\Subsubsection` commands.

A section is then referenced by writing

```
In \Ref{Section{do} we do tell you what to do.
```

The `\Ref`command supports both backward and *forward* references. The main advantages of the `labels` package over the standard `\label` and `\ref` mechanisms are:

Type safety Writing `\Ref{Figure}{do}` if `do` was a label assigned to a section.

Common abbreviations The `\Ref` macro is set to use common abbreviations. Thus, if you write

```
\Ref{Section}{do}
```

in your input, the output will be:

Sec. 2 …

Similarly `\Ref{Figure}{xy}` might come out as

Fig. 14 …

Clear mindedness No need to remember to put the ~ before the `\ref` command.

The `labels` package is useful for environments as well:

```
\begin[boats]{Figure}{Boats used to cross the Atlantic}
    ...
\end{Figure}
```

Will produce
And then you can give reference to the figure using the `\Ref` macro

```
In \Ref{Figure}{boats} we see the ...
```

Will produce the output
Figure:boats
In Fig. 6.1 we see the …

Similar trick works for tables, which are defined using the `Table` environment which takes two arguments, the label, and the caption.

The `\newRef` command can be used for changing the format of references. The following are predefined in `labels.sty`, but can be overridden by the user.

```
\newRef{Lemma}{Lem.~\ref{Lemma:#1}}
\newRef{Theorem}{Thm.~\ref{Theorem:#1}}
\newRef{Definition}{Def.~\ref{Definition:#1}}
\newRef{Corollary}{Corr.~\ref{Corollary:#1}}
\newRef{Proposition}{Prop.~\ref{Corollary:#1}}
\newRef{Chapter}{Chap.~\ref{Chapter:#1}}
\newRef{Section}{Sec.~\ref{Section:#1}}
\newRef{Appendix }{App.~\ref{Appendix:#1}}
\newRef{Table}{Tab.~\ref{Table:#1}}
\newRef{Figure}{Fig.~\ref{Figure:#1}}
\newRef{Algorithm}{Alg.~\ref{Algorithm:#1}}
```

7 The sentence structure

7.1 The Meat and the Bones

Every scientific sentence has words of four kinds in it:

...

Figure 6.1: Boats used to cross the Atlantic. `Figure:boats`

Bones (or Objects) These are terms and nouns defining the entities of the domain of discourse. Most sentences of the paper shall discuss these objects, each contributing, in one way or another, additional information to the reader on these entities. The bones should be defined clearly and formally at the beginning of the paper, in order to avoid repetitive qualifications in the sequel. The beginning of the paper should define the bones; this would save repetitive, semi-definitions in the sequel.

In order to avoid repetitive Clear and formal definitions of the bones at the beginning of the paper eliminates repetitive and tedious inaccurate references in the sequel. Definitions of bones at the beginning of the paper, if they are done in a clear and accurate manner, would save you tedious and vague references to the bones. An up front definition of terms saves tedious, vague and... Formal and clear ... Repetitive, If you find yourself referring to the same term again and again, Repetitive references can be avoided by an up front clear definition of the bones.

Define the bones at the beginning. This will save you the tedious work of making repetitive qualifications which are invariably vague.

Use either the `\emph` macro, or a proper `\begin{definition}` ...`\end{definition}`. Sometimes, discovering the bones and is not easy, and is only done in the midst of your writing; you hit a bone when you find yourself using the same phrase again and again.

Meat (or Attributes) Each sentence should reveal more information about the bones. The common way of doing so is by using attributes. Attributes supply the valuable information regarding the objects, and should have the main role in sentences. In choosing attributes, use a rich and accurate language, as oppose to the simple language that should be used when describing the objects. Use the most specific word. If a word has a mundane usage, it is not very specific.

When using the same term or verb (meat) several times, make sure that the meaning remains the same. Only in rare cases the meaning can be different, but in these cases the differences must be made clear to the reader.

Connecting tissue These are noise words such as “the”, “are”, “hence”, “such”, etc. There are two kinds of connecting words: those which are used to bind together the nouns and verbs make it grammatically correct and expressive of some small coherent idea. The other kind of connecting words, are those used in connecting each sentence to the previous one, and, less frequently, in leading to the next sentence. These are noise words such as “usually”, “moreover”, “hence”, “such”, “therefore”, “however”, etc. They may be added at the beginning, somewhere in the middle or the end of the sentence.

Makeup This are fancy words, which usually add nothing to the content. Avoid these as much as you can.

7.2 Tips

Here is a list of rules and tips which I collected while working with many of my students:

1. Conjugations associate to the left; hyphens denote multiple conjugations, as in: “pre- and post-conditions are the essential ingredients of *programming by contract*”.

2. “Few” means several. “A few” means a small number.

3. New terms must be *emphasized*. Frequently, a term is emphasized twice: When first mentioned, and again when defined in greater detail. The author may choose which one of these occurrences to emphasize, when they are close to each other. When these two occurrences are near, it is common to emphasize only one of them.

4. The first use of a term, defined in some other work, must by accompanied with proper reference to this work.
5. Sentences beginning with “this”, “that”, “it” are usually ambiguous.
6. Use symmetry to clarify structure. Asymmetry confuses the reader.
7. If you use the same term (object) several times, then the meaning must be the same. In the rare cases that the meaning differs, care must be taken to highlight the change in semantics.
8. Cohesiveness: always related to the previous text and lead to the following text.
9. Each idea must be asserted and proved once and only once. It is acceptable to make future references and backward references, but these references must not repeat the main argument or the definition. Never repeat the arguments at the same level of detail.
10. Bones: Sometimes the cure to awkward, verbose and cumbersome writing is in simply defining a new term.
11. If you cannot write a sentence, start over with a different point or subject.
12. Every sentence must end with its strongest part. Every paragraph must end with its conclusion.
13. Good use of fonts makes it easier on the reader.
14. Does “thus” mean follows?
15. Clearly, obviously, simply, etc. are usually not so clear, obvious, and simple.
16. Don’t use quotes and . . . to cover for sloppy writing and imprecise thinking.
17. Don’t use slash (/) as a punctuation mark.
18. Do not use synonyms adjectives and adverbs in the same sentence.
19. Question every “very”.
20. Techniques: go from passive to active.
21. Techniques: Connect sentences and break them.
22. Problem indicator: the same word is used at the ending of one sentence and in the beginning of the following one.
23. Remove redundant words.
24. Identify the individual (instance) and the group (class) of each noun. If it is existential quantifier, use “a” or “an”. If it is a specific element (a given, or a constant in first order logic), use “the”. If it is a singleton object, don’t use “a” or “the”.
25. Connect sentences properly. Use “which” and “that”.
26. If you write “frequently”, “many”, “typically”, “mostly”, you appear to unassertive.

8 Discussion of a Home Assignment of a Student

nment

In this section we bring various comments on a student's home assignment, and some LaTeX concepts related to it.

8.1 Title

The title is

"Scientific Writing - Exercise 1"

where it had been nicer written as

Scientific Writing
Exercise 1

or

Scientific Writing
Assignment 1

8.2 General comments

- List the name of the paper referred to in the bibliography.
- Referring to the authors of a paper as *the authors* is considered to be disrespecting.
- A word is referred to as a *noise word* when it does not add information. *Actually* is usually such a word. Noise words should be avoided.
- A generally hard problem is having the subject well defined in each sentence. A sentence starting with words such as *that*, *this* or *it*, is often ambiguous. Usually words like these ones are agreed to refer to the subject of the former sentence. When reffering to the same object in a few sentences, consider giving it a *name*. Using names usually makes sentences clearer.
- Every sentence should be put so that it could have ended with an exclamation mark. A sentence should end with it's strongest part.
- Sentences can be connected using different connection words:
  - words describing a logical sequence, such as *therefore*, *consequently*, *subsequently* or *thus*
  - words describing contrast, such as *nonetheless*, *nevertheless* and *however*
  - phrases of example, such as *for example* and *for instance*

  Nevertheless, connection words tend to clog up the paper. A colon (`:`) is one way to avoid connection words.

8.3 Examples

nment

In this section we examine a few sentences drawn home assignment, and offer corrections and rephrasing.

In the paper entitled "Efficient ... " In *Efficient* .....

"*It is important to specify the institutes of which the authors come from.*"

1. *Affiliations* is more suitable here than *institutes.*
2. *Important* is too general and daily. *Essential* could be used.
3. A detail missing here is the *reason* why the affiliation should be specified: one reason is that they provide the funding.
4. An alternative sentence could be: *Authors specify their affiliations for funding, and readers are more likely to be appreciative of papers written in honorable institutes.*

"*The order in which authors appear is important.*"

1. The word *important* appears both in this sentence and in the former one, it should be replaced.
2. The word *appear* does not add information: use of *ordering* could shorten the sentence.
3. The word *used* appears too often, and is too daily.
4. Two other words that could match here are *naming* and *referring*.
5. This sentence does not belong in this paragraph!
6. An alternative sentence could be: *When a non-alphabetical order is used...*

"*The abstract is written in a unique, specific language.*"

1. The difference between the words *unique* and *specific* is unclear, one of them is redundant.
2. An alternative could be: *The abstract summarizes the paper in the domain's jargon.*

"*Usually this part is searched by search-engines. For this purpose, the abstract can include a part of keywords of the paper.*"

1. The word *part* appears twice, having a different meaning in each place. Both meanings are unclear.
2. In *for this purpose* it is unclear what the purpose is.

"*When a paper is given to a conference, it is judged first according to it's abstract.*"

1. The word *given* is too simple. *Submitted* is more appropriate.
2. Instead of *according to*, write *by*.
3. "it's" is a shorthand for "it is". The gender neutral possessive form "its" should have been used here.

"*In English conjugations associates to the left. Use hyphens to make your point clear.*"

1. The sentence can be shortened: *"Conjunctions associate to the left; hyphens add clarity"*.

   Note that now it is unclear. We can add an example.

2. Another alternative: *Hyphens denote multiple conjunctions, as in:* (and here comes an example).

"Every new term must be emphasized*. Frequently, a term is emphasized twice: When it is first mentioned,. . . "*

1. Instead of *Every new term* we could shorten: *New terms.*

2. Instead of *When it is first mentioned*: *When first mentioned*.

   Generally, verbs and adjectives can be replaced. Try to form sentences shorter and emphasize the main part.

"The first mentioning of a term which was defined by other authors, must include reference to this other work." This sentence is not very impressive. It could have been rephrased as:

1. *Include reference when first using . . . .*

   But this way the strong point comes in the beginning of the sentence.

2. *The first use of a term, defined in some other work, must be accompanied with proper reference to this work.*

'Only in rare cases the meaning can be different, but in these cases the differences must be made clear to the reader"

The sentences could be rewritten as: *In the rare cases that the meaning can be different, the difference in semantics must be highlighted. .*

9 Rephrasing a Sentence–Examples

1. In sally:00 [?] we found the paragraph:

   > It is important to specify the institutes of which the authors come from. The order in which authors appear is important: usually alphabetical order is used. Using a different order emphasizes the author listed first. When naming the authors, usually use last names only. Writing full names is a very special honor. Referring to the authors of a paper as *the authors* is considered to be disrespecting.

   By reviewing each sentence separately we get the following points:

   - "It is important to specify the institutes of which the authors come from." – The predicate of this sentence is *authors* and the two attributes are *institutes* and *the importance to specify it*. Instead of using these attributes we could write *affiliation*. It is best to avoid the word *from* at the end of the sentence. The word *important* is not well used in this context; it's meaning gets blurred. A different word that would give a more specific meaning is *essential*.

     One way to rephrase this sentence is: "Author's affiliation is essential". This proposal lacks the reason for including the affiliation. From the author point of view, affiliation of a prestige institute credits him. On the other hand, from the institute outlook, it's a gratitude for supplying the funding. Putting it all together results in: "Author's affiliation is essential. Authors will specify affiliation for funding purposes. Readers are more likely to accept authors with important affiliation."

- "The order in which authors appear is important: usually alphabetical order is used." – We have here a repetition of the word *important* from the previous sentence. It can be replaced with: "Ordering of authors name is significant." Words such as: *usually*, *used* are too common and should not be abused.
- "Using a different order emphasizes the author listed first."– This sentence should be joined together with the previous one e.g., "Referring to the authors in a non-alphabetical order emphasizes the importance of the first."

2. Another paragraph found in [?] is: sally:00

   > The abstract is written in a unique, specific language. It summarizes the paper. Usually this part is searched by search-engines.

   By doing the same kind of review as in the previous example we find:

   - The difference between the words *unique* and *specific* isn't clear. In this case it would have been better to omit one of them.
   - The word *it* in the second sentence refers to the subject of the first one, although this is not clear because the sentence has more than one noun. In these cases, starting it with an *it* can be confusing.
   - A better way to state the same idea would have been by combining the first two sentences: "The abstract summarizes the paper using a unique language."
   - The third sentence introduces a word which wasn't discussed before (*part*) and can be inconclusive to the reader.

3. The sentence, "When a paper is given to a conference, it is judged first according to it's abstract.", could be rewritten using more suitable verbs: "When a paper is submitted to a conference, it is judged first by it's abstract."

4. "In English conjugations associates to the left. Use hyphens to make your point clear."

   A more suitable way to say the same would be: "Hyphens add clarity." It is recommended to bring an example to illustrate your point: "Hyphens denote multiple conjugations, as in: pre- and post-condition are essential ingredients of *programing by contract*."

5. "Every new term must be *emphasized*. Frequently, a term is emphasized twice: When it is first mentioned, and then again when a more complete definition is given to it." [?] yogi:00

   It is best to write succinct sentences so that redundant words such as *it is* in this example could be removed.

6. "If these two occurrences are close, the author may choose which one of these to emphasize." [?] yogi:00

   When writing a sentence properly, there exists an innate conflict. We prefer to use active voice and not passive, forcing the subject to appear at the beginning, but we also would like the essence to appear at the end (to emphasize it). We can rephrase this example in the following way: "The authors may choose which one of these occurrences to emphasize, when they are too close to each other." In this way we emphasize with active voice. Alternatively we could write: "When these two occurrences are near, it is common to emphasize one of them." – putting the essence last.

7. “The first mentioning of a term which was defined by other authors, must include reference to this other work.” [yogi:00] [?]

1. In [sally:00] [?] we find the following paragraph:

   > It is important to specify the institutes of which the authors come from. The order in which authors appear is important: usually alphabetical order is used. Using a different order emphasizes the author listed first. When naming the authors, usually use last names only. Writing full names is a very special honor. Referring to the authors of a paper as *the authors* is considered to be disrespecting.

   Lets review each:

   - “It is important to specify the institutes of which the authors come from.” – The predicate of this sentence is *authors* and the two attributes are *institutes* and *the importance to specify it*. Instead of using these attributes we could write *affiliation*. The sentence ends with a preposition (*from*), which is best to avoid. The word *important* is so widely used that it’s meaning is blurred. A different word that would give more specific meaning is *essential*.

     One way to rephrase this sentence is: “Author’s affiliation is essential”. This proposal lacks the reason for including the affiliation. From the author point of view, affiliation of a prestige institute gives him credits, and from the institute outlook, it’s a gratitude for supplying the funding. Putting it all together results in: “Author’s affiliation is essential. Authors will specify affiliation for funding purposes. Readers are more likely to be accepted authors with important affiliation.”
   - “The order in which authors appear is important: usually alphabetical order is used.” – We have here a repetition of the word *important* from the previous sentence. It can be replace as follows: “Ordering of authors name is significant.” Words such as: *usually*, *used* are too common and should be limited.
   - “Using a different order emphasizes the author listed first.”– This sentence should be joined together with the previous one e.g., “Referring to the authors in a non-alphabetical order emphasizes the importance of the first.”

2. Lets consider the following sentence from [sally:00] [?]:

   > The abstract is written in a unique, specific language. It summarizes the paper. Usually this part is searched by search-engines.

   The difference between *unique* and *specific* isn’t clear.

   It, in the second sentence, must refer to the subject of the first one. When a sentence have more than one noun, starting it with an *it* can be confusing. Therefore combining these two sentences together is preferred: “The abstract summarizes the paper using a unique language.”

   The third sentence introduces a word which wasn’t discussed before (*part*) and can be inconclusive to the reader.

3. Another example from [sally:00] [?]:

   > When a paper is given to a conference, it is judged first according to it’s abstract.

Using more suitable verbs, the sentence can be rewritten as: “When a paper is submitted to a conference, it is judged first by it’s abstract.”

4. Let us review the following example from [?]:

   > In English conjugations associates to the left. Use hyphens to make your point clear.

   More suitable way to say the same is: “Hyphens add clarity.”

   It is recommended to bring an example to illustrate your point: “Hyphens denote multiple conjugations, as in: pre- and post-condition are essential ingredients of *programing by contract*.”

5. Another example form [?]:

   > Every new term must be *emphasized*. Frequently, a term is emphasized twice: When it is first mentioned, and then again when a more complete definition is given to it.

   It is best to write succinct sentences so redundant word such as, *it is* in this example should be removed.

6. In [?] we also find:

   > If these two occurrences are close, the author may choose which one of these to emphasize.

   There is a conflict in writing a proper sentence. We want it to be active and not passive – which force the subject to appear at the beginning, but we also want the essence to appear at the end. We can rephrase this example as follows, while emphasizing on active form: “The authors may choose which one of these occurrences to emphasize, when they are too close to each other.” or we can write: “When these two occurrences are near, it is common to emphasize one of them.”–placing the essence last.

7. Finally from [?]:

   > The first mentioning of a term which was defined by other authors, must include reference to this other work.

   Sometimes changing the sentence from passive to active does more harm than good. Rephrases such as, “Include reference when first using a term defined elsewhere.”” or “When using a term. . . ”, although in active form, are worse.

   In this case we’ll prefer leaving the essence of the sentence at the end, but use a better choice of words: “The first use of a term, defined in some other work, must be accompanied with proper reference to this work.”

10 Discussion of a Home Assignment of a Student

In this section we bring various comments on a student’s home assignment, and some LaTeX concepts related to it.

10.1 Title

The title is

"Scientific Writing - Exercise 1"

where it had been nicer written as

Scientific Writing
Exercise 1

or

Scientific Writing
Assignment 1

10.2 General comments

- List the name of the paper referred to in the bibliography.
- Referring to the authors of a paper as *the authors* is considered to be disrespecting.
- A word is referred to as a *noise word* when it does not add information. *Actually* is usually such a word. Noise words should be avoided.
- A generally hard problem is having the subject well defined in each sentence. A sentence starting with words such as *that*, *this* or *it*, is often ambiguous. Usually words like these ones are agreed to refer to the subject of the former sentence. When reffering to the same object in a few sentences, consider giving it a *name*. Using names usually makes sentences clearer.
- Every sentence should be put so that it could have ended with an exclamation mark. A sentence should end with it's strongest part.
- Sentences can be connected using different connection words:
  - words describing a logical sequence, such as *therefore*, *consequently*, *subsequently* or *thus*
  - words describing contrast, such as *nonetheless*, *nevertheless* and *however*
  - phrases of example, such as *for example* and *for instance*

  Nevertheless, connection words tend to clog up the paper. A colon (`:`) is one way to avoid connection words.

11 Tables and Figures

gures

Objects such as figures and tables may be too large to be placed at the current position in the text or even on the current page; they are sometimes floated forward to a place where they can be typeset.

There are two environments in LaTeX for figures and for tables separately, which are responsible for floating the object to the appropriate place.

```
\begin{object}[pref]
 The body of the object.
\caption{Title}
\label{Name}
\end{object}
```

where `object` is either `table` or `figure`. The command `\caption` may appear in every place between `\begin{object}` ... `\end{object}` environment and according to its location the title will be either above or under the object.

Together with `\label{Name}`, the command `\ref{Name}` creates the object number (separately for figures and tables).

By means of an optional argument `pref` (in the square braces) user can define object's placement in the text. `pref` is a sequence of one or more letters from the following list (with a preference from left to right):

t place the object at the top of the page;

b place the object at the bottom of the page;

p place the object on the separate page, entirely includes "floated bodies" only;

h place the object neatly in the current text, without floating it.

In addition, there are the rules to determine where an object (table or figure) will appear in the text:

- The object is put at the earliest place that does not contradict the optional argument.
- The floated body cannot be before its environment definition and also cannot appear before the previously defined floated body from the same kind (figure or table).
- Putting the object on the current page cannot overfull it.
- If the sign `!` appears in square braces before the letters sequence of the optional argument, then the LaTeX operates accordingly to the sequence, ignoring the above rules.

The default optional argument is `tbp` sequence.

11.1 Tables

ables The basic idea of the article need to be understandable from the tables of this article. In a book and a thesis there is a list of numbered tables, which is automatically generated by the command `\listoftables`.

The principles of table producing:

1. The title of the table should be meaningful and independent on text.
2. The title(caption) of the table need to be independent on columns and rows.
3. It is not allowed to demand from the reader to perform some arithmetical operation to understand the table content.

4. The definition of units and measurement conditions, that appears in the table must be exact.
5. Unnecessary inefficient elements are not to be shown in the table.
6. The data is displayed in reasonable precision (not exaggerated).
7. The title of the table should be centered.
8. The numbers are justified to the right.
9. It is important to reduce the tables and to compress the including material up to the possible limit without loss of information.

Earlier we became familiar with the `table` environment which makes possible to entitle, to label, to refer to it and to place the table in the text. Now we consider the environment which, in fact, creates the table. Such environment is called `tabular` and has a structure:

```
\begin{tabular}{format}
...
\end{tabular}
```

The `format`, in the simplest case, is a sequence of letters, that describes the structure of the table columns (one letter for each column). These letters may be as follows:

l means a left aligned column;

r means a right aligned column;

c means a column with centered text.

Between the commands `\begin{tabular}{format}` and `\end{tabular}` the text of table is located. Within the command `\\` separates the rows of the table, and the sign & separates the columns of the table inside one row.

There exists a possibility to create a ruled tables in LaTeX. The horizontal lines are defined by the command `\hline` and have a width which is equal to the general table width. To draw the vertical lines along the general table height we may put the symbol `|` between the letters in the `format` sequence.

To locate the table at the center (by default it is pinned to the left of the page) we use the following commands:

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

or

```
\centering
```

To create a more complicated table, such as one with inscription that includes a number of common columns, in place of the corresponding table column we need to use a command

```
\multicolumn{n}{format}{text}.
```

This command has a three necessary arguments:

- `n` is a number of columns that this “not standard” column includes.

- `format` (like `\begin{tabular}{format}`) contains sequence of letters `l`, `c`, `r`, and/or columns separator – the symbol `|`.
- `text` is a text for this column.

For drawing horizontal line, which include a certain number of columns we need to use the command `\cline{clmn1-clmn2}` after a `\\`. As in our example bellow, `\cline{2-3}` makes a horizontal line under second and third columns.

	Grades	
Name	Hedva	Algebra
Cohen Uri	80	85
Levi Gil	74	79
…	…	…

```
\begin{center}
  \begin{tabular}{|l||c|c|}
  \hline & \multicolumn{2}{c|}{Grades} \\
           \cline{2-3} Name & Hedva & Algebra
  \\ \hline\hline
  Cohen Uri & 80 & 85 \\ Levi Gil & 74 & 79 \\
  \ldots    & \ldots & \ldots \\ \hline
  \end{tabular}
\end{center}
```

As we have mentioned above, if in `table` environment we write `\label{tab:word}`, so at every place in the text we could refer to this table by means of `\ref{tab:word}`. For example,

```
From the Table \ref{tab:word} we
can conclude that the most frequent
English word is \textbf{the}.
```

The output we obtain

From the Table tab:word 11.1 we can conclude that the most frequent English word is **the**.

Word	Frequency (%)	Word	Frequency (%)
the	6.421	that	1.244
of	4.028	is	1.034
and	3.150	i	0.945
to	2.367	it	0.930
a	2.029	for	0.770
in	1.778	as	0.764

Table 11.1: The usage frequency of some English words. tab:wo

The following LaTeX code produces the above table:

```
\begin{table}[htbp]
  \begin{center}
     \begin{tabular}{|l|c||l|c|}
        \hline
        Word    & Frequency (\%) & Word  & Frequency (\%) \\
        \hline \hline
        the     &    6.421       & that  &     1.244      \\
        of      &    4.028       & is    &     1.034      \\
        and     &    3.150       & i     &     0.945      \\
        to      &    2.367       & it    &     0.930      \\
        a       &    2.029       & for   &     0.770      \\
        in      &    1.778       & as    &     0.764      \\
        \hline
     \end{tabular}
  \caption{The usage frequency of some English words.}
  \label{tab:word}
  \end{center}
\end{table}
```

11.2 Figures

c:fig

The figures (graphs) may be of different types. Some of such types are considered below. As well as for tables we can make a list of numbered figures, which is automatically generated by the command `\listoffigures`.

Categories of graphs:

1. Graphs may be conceptual, those which represent their idea without using a visual and defined language, for instance Visio.

2. Graphs, which use visual or well-known language, such as diagrams (Rose, VML).

3. Graphs which represent some data, for example, function plots, statistical data or tables. This is another way to create tables, e.g., with help of Excel program (Figure 11.2).

fig:graph

4. Images and photographs.

Description of the figure that appears in caption must be such that anyone can understand it without considering the graph itself.

If we want to include some picture or graph into LaTeX document we need use the following command

```
\includegraphics[key = value, ...]{fname.eps}.
```

Here the graphics file need to be in Encapsulated PostScript (EPS) format. In addition, in preamble we must include the package

```
\usepackage[driver]{graphicx}
```

where the `driver` is a program, which is used to convert from dvi to PostScript. For instance, a `driver` we use is `dvips` program.

Figure 11.1: The mid-term exam results of the course ”Set Theory” for the previous (left column) and the current (right column) semesters.

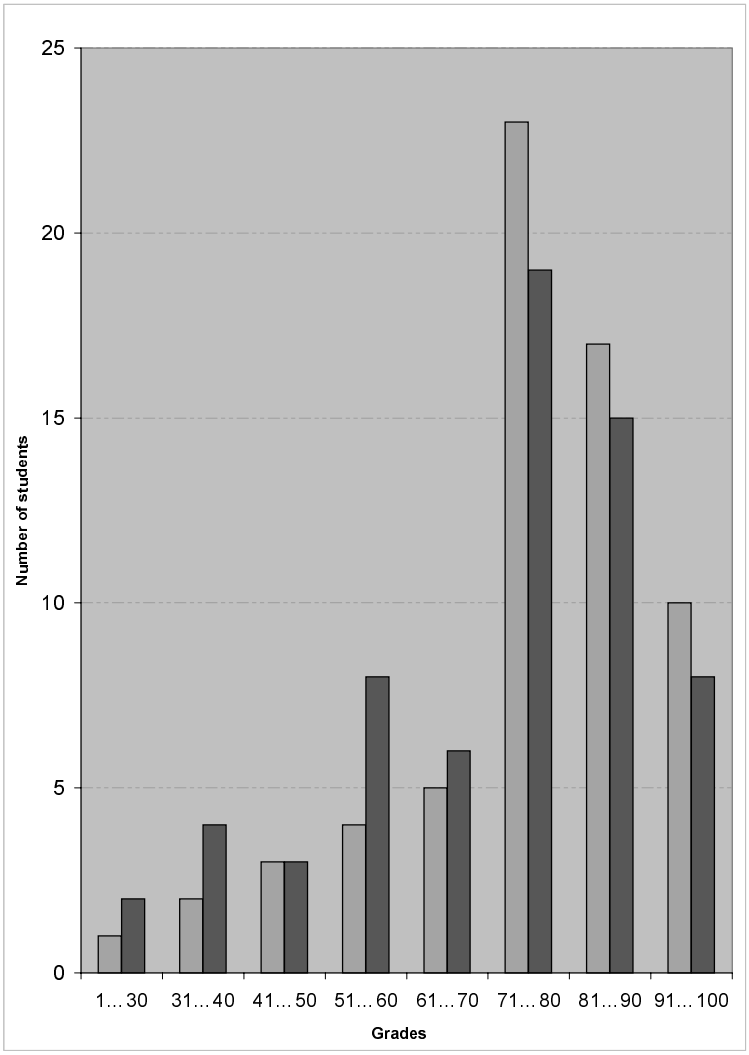

fig:graph

Let’s return to the command `\includegraphics` as defined above. The optional argument in square braces is a list of keys (such as `width`, `height`, `angle`, `scale`, ...) with their corresponding values. This makes it possible to scale, to cut out a necessary part of picture, to rotate it by the certain angle (given in degrees) or to display it in a certain size (in `mm`, `cm`, `in`, `pt`, ...). The details can be found in [1, 2].

Grif, Lamp

The examples of using this command with the `figure` environment, discussed in the Section ??, are follow.

ssec:tab

```
\begin{figure}[htbp]
   \begin{center}
   \includegraphics[width = 7cm, height = 6.5cm]{clown.eps}
   \end{center}
   \caption{The clown.}
   \label{fig:clown}
\end{figure}
```

In the Figure 11.2 the sad clown is shown, and in the Figure 11.3 plot of some function is displayed.

fig:clown

fig:plot

Figure 11.2: The clown. fig:cl

```
\begin{figure}[!htb]
   \centering
   \includegraphics[scale = .4]{plot.eps}
   \caption{Plot of function.}
   \label{fig:plot}
\end{figure}
```

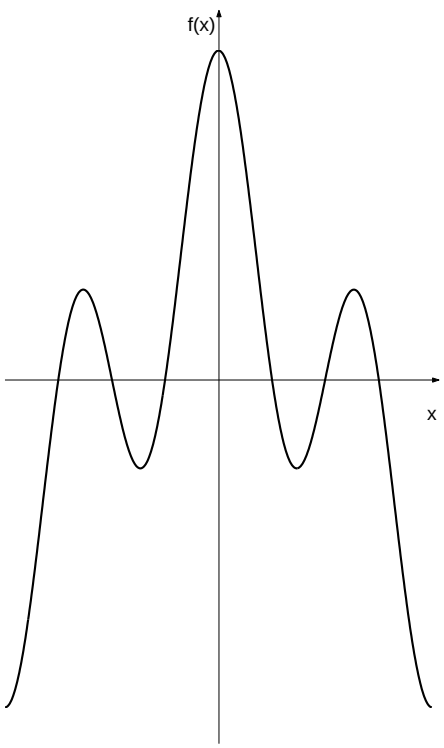

Figure 11.3: Plot of function. fig:pl

12 Advanced topics in LaTeX

12.1 Numbering

ering

This section discusses numbering. Explanations are given for figures numbering, and are applicable to tables or sections numbering, as well.

Figures are numbered, and their numbers appear in their titles, and in other places in the text. Hand typing of expressions such as *in Figure 2* or *in Section 3* is not recommended: changing the style of entire document (such as changing *in Figure. 2* to *in Fig. 2* throughout the paper), or removing one figure, shall require many changes.

LaTeX has an automatic numbering mechanism. Figures, tables and sections all have separate internal counters, called `thesection`, `thefigure` and `thetabel`. The counters are incremented after declaring the object (for a section)—or, for floating bodies (figures, tables)—after the `begin` and `caption` commands.

For example, after the two commands

```
\begin{figure}
\caption{some caption}
```

the counter *thefigure* is incremented.

When referring figure's number, give it a label, using the command

```
\label{some label name}
```

The overall series of commands should be:

```
\begin{figure}
\caption{some caption}
\label{some label name}
\end{figure}
```

After the *label* command, the label given refers to the value of the counter (in this case, `thefigure`). The command

```
\ref{some label name}
```

will print the value of the counter, which is the figure number.
In this place it is appropriate to use a tilde (`~`) (*a non breakable space*)—for example, we could write

```
In Section~\ref{some label name}
```

in order to get:

In Section 3

LaTeX will not break the expression into two different lines.

Using a Macro
Instead of typing the mentioned text each time, define a *macro* once and use it later.
For example, to define a macro named *Ref*, with two parameters, in TEX:

```
\def\Ref#2{#1~\ref{#1:#2}}
```

and in LATEX:

```
\newcommand{\Ref}[2]{#1~\ref{#1:#2}}
```

where *#1* refers to the first parameter, and *#2* — to the second one.
In LATEX, Defining the same macro twice will produce an error message, and so defining macros in LATEX is prefered.

12.2 LATEX's auxiliary files

files

TEX's input consists of a *.tex* file, and macro files. Its output includes a few output files, of which the main one is a *.dvi*[7] file. The *.dvi* file can be viewed and turned into a postscript file.

TEX also outputs auxiliary files, which include the *.log* and *.aux* files. The *.log* file contains different comments concerning the compilation, including error messages. The *.aux* file contains data to be used during the next compilation, such as information needed to handle commands like `\ref`. During the first run, the compiler saves all required data, and in the second run it can place all text correctly. The compiler also checks that the input *.aux* file (if exists) is identical to the one produced. If this is not the case (for example, if the file contained circular definitions) and error message will be produced.

TEX finds the macro files using the environment variable of Unix `TEXINPUTS`. If the paths are `p1`, `p2` and `p3`, the variable is set by using the line:

```
TEXINPUTS p1; p2; p3
```

Saving the macro files on the local disk is a mistake, since they can be changed.

12.3 Building a Bibliography

raphy

The bibliography is one of the main parts of a paper: the paper is partially judged by it. Readers check if it includes important work done in the subject.

In order to create a bibliography, use database files, which contain data concerning the papers to be mentioned in the paper (usually large, standard files are used, and papers are added to them if necessary). For each paper in the database a *key* is defined. In order to refer to a paper, refer to it's key, using the command

```
\cite{the paper's key}
```

This command will print data relevant to the paper.

Creating a bibliography is done by the following commands

```
\bibliographystyle{abbrv}
\bibliography{main}
\eject
```

where the first command sets the bibliography style (stype *abbrv* is usually used), the second one creates the bibliography using the database file (here it is *main*), and the third one creates the bibliography.

The bibliography includes all the papers we have referred to in the paper (using the `\cite` command).

[7] *Device Independent Format*

Behind the Scenes of creating the bibliography
When TEX runs first, it lists the references to the bibliography database in the *.aux* file. BibTeX is a program that searches the *.aux* file for references to the bibliography database, and using this information, creates a *.bbl* file which contains data needed to create a bibliography list for the paper. In the second run of the compiler, BibTeX will create the *.bbl* file. In the third run, if the command `\bibliography{file name}` appears, the bibliography is created using the *.bbl* file.

12.4 Creating a Table of Contents

tents

A *table of contents* is a list of sections, along with their page numbers. A table of contents is included in very long papers (over 70 pages), and in thesises, at the begining of the paper.

To create it, use the command `\tableofcontents`. Similar commands are `\listoffigures` and `\listoftables`. When using these commands, TEX creates *.toc*, *.lot* and *.lof* files, which are used in the next run, to create the table of contents or lists of figures and tables.

13 Mathematical Formulae

We write mathematical formulae and symbols inside the `math` environment or one of its variants (such as `equation`). Because mathematical text is so frequent in scientific writing, TEX provides many shortcuts and mechanisms for entering and exiting this environment. Mathematical formulae are either inlined or displayed. Inlined formulae are written between dollar symbols. Displayed formulae are enclosed between `\[` and `\]`, or between double dollar symbols, etc.

There are many alternative ways for including formulae in the text: inlined formulae can also enclosed between `\(` and `\)`; displayed numbered formulae are written in an `equation` environment; also, the `amsmath` package adds useful environments for aligned displayed equations.

13.1 Subscripts and Superscripts

The underscore character (_) denotes subscripts, while ^ denotes superscripts. Here is an example:

```
Compare~$\sum_{i=1}^(2i-1) = n^2$ with \[
   \sum_{i=1}^(2i-1) = n^2.
\]
```

Compare $\sum_{i=1}^{n}(2i-1) = n^2$ with

$$\sum_{i=1}^{n}(2i-1) = n^2.$$

Sometimes, we would like to write plain text using superscript or subscript, for example, to add a “th” as a superscript to a number. We can go into the math environment and write, for example, `$20^{th}$` which would indeed appear as 20^{th}, but sometimes we would rather not. An acceptable solution is to use

```
\textsuperscript{textit{th}}
```

which would yield correct results but is cumbersome to use. The best solution is to define a macro:

```
\newcommand{\nth}[2]{#1\textsuperscript{\textit{#2}}}
```

which would appear as 20^{th} when we write `\nth{20}{th}`.

13.2 Mathematical Symbols

LaTeXcommands exist for most common mathematical symbols. Other symbols can be found in other packages. Most of these commands only work within the math environment.

For example, the code

```
Let $c\in{C}$ be a class of type $t\in{T}$
```

would appear as:

> Let $c \in C$ be a class of type $t \in T$

13.3 Typefaces for Mathematical Formulae

It is common that mathematical formulae appear in *italic*. This is applied automatically to everything in the math environment.

In regular text, some pairs of letters, such as ff and fi are brought together and take a special form called *ligature* which consume less space. The most famous examples are the *ff* and *fi*, other ligatures are *ffi*. In mathematical environments, ligatures are automatically disabled to make it clear that that fi is the product of f by i.

This false-multiplication problem occurs for other mathematical symbols. If everything was typefaced the same, then writing $log(n)$ would have look as the multiplication $l * o * g * (n)$. To prevent this, we need to use the appropriate command, so we would write `\log(n)` instead of `log(n)` resulting in $\log(n)$ instead of $log(n)$. In the case of a logarithm of base 2, we use the `\lg` command instead.

14 Phrasing Definitions and Theorems

The following are some rules-of-thumb, to be used when phrasing definitions and theorems.

1. In a text with as $M_1, M_2, \ldots$, where it is clear that M denotes a method, then it is recommended not to write *Methods* $M_1, M_2, \ldots$ unless this is the beginning of a sentence.

2. If a variable or a symbol appears only once in the definition, it is likely that there is a problem with the definition. For example, we can shorten "Let $M_1, M_2, \ldots$ be the methods of a class C_1" to "Let $M_1, M_2, \ldots$ be the methods of a class", which does not confuse the reader with superfluous symbols.

3. A formula is a valid part of the syntactic structure of the sentence, and therefore should be surrounded by punctuations as usual.

4. It is common to begin lines in mathematical texts with words such as *Let* and *Then*.

14.1 Escaping the Mathematical Special Characters

Carets are extremely rare in ordinary text. If the need arises, a caret can inserted using the `\verb` command: `\verb+^+`, or by its ASCII code: `\symbol{94}`. (Note that the `\^` command places a caret over the next character, as in many European languages.)

The dollar symbol is more common in English text, and can be inserted as `\$`. Underscores are abundant in computer related texts. Again, the `\_` command works.

14.2 The Programming Metaphor for Mathematical Notation

A consistent, coherent and clear use of mathematics is often the result of thinking of your paper as if it was a program in some statically typed programming language. Single letter notations such as i, ϵ and G are scalar variables. Subscripts and functional notation such as x_i, $g(y)$ are array variables. Sometime superscripts are used for record structures, as in ℓ^u and ℓ^v.

This metaphor helps remembering that every new mathematical variable must be defined it is to be used. In some cases the definition is ad-hoc, much like the use of a temporary variable, and frequently shortly *after* use. In the following, the function g and the value a must have been previously defined.

> We therefore have that $\Leftrightarrow a \leq g(x) \leq a$ for $0 \leq x < 1$.

In contrast, the variable x is defined after it has been used. The scope of this definition is limited to the sentence itself, unless intentionally stretched as in:

> For such x, $g(x^2) < g(x^3)$ also holds.

Unless specifically restricted, a mathematical definition has global scope.

- A *hierarchy* is a pair of a set T of types, and a partial order relation $\prec$.
- Let C be a graph.
- Henceforth, c will denote an arbitrary class.

The latter example is a case in which the global scope of the definition of C is made explicit.

It is a good practice to remind the reader of the type of a variable whenever it is at the first level of the text. Adhering to this practice, and extending it when appropriate to complex expressions, will help you to keep the rule that sentences should never start with a formula.

- Node n is in the connected component of the graph.
- Condition $t' \prec t$ holds since t' was found in a breadth first search starting at t.

It is very common in mathematical writing to implicitly adopt a FORTRAN-like convention in which designated sections of the mathematical symbols vocabulary are reserved for denoting variables of a certain type. In this convention, the type of most variables is determined by their type. Some of the sections typically used in this convention are:

1. $a, b, c, \ldots$ (lower case letters from the beginning of the English alphabet).
2. i, j, k and l (often also ℓ), typically used as integer indices.
3. m, n, p, q, r, and s, typically used to denote integers. The symbol o is rarely every used for anything other than denoting the origin.
4. x, y, z, u, v, w (lower case letters from the beginning of the English alphabet).
5. $A, B, C, \ldots$ (lower case letters from the beginning of the English alphabet).
6. α, β, γ (and rarely δ and ϵ).
7. $\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$ (calligraphic letters).
8. $\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots$ (mathematical bold face letters).

This typing convention is often implicit. In cases where heavy notation must be used, notably in parsing theory, the convention is explicitly described.

Avoid using obscure mathematical characters such as ϱ, ς, $\Im$, and $\wp$ whose name would be a mystery to most readers. An exception is when these are an established convention of the domain of discourse. For example, $\Im(z)$ is often used to denote the imaginary part of a complex number z, and $\wp(s)$ is often an established notation for the power set of a set s.

Good notation is hard to come by—even experienced authors tend to change their notation as they write their paper. Does n denote the input size, or does it stand for a node in a graph? Does p denote a prime number, or is it a processor in a distributed computing environment?

By using meaningful names instead of the terse mathematical notation, changes can later be made. The preamble is a great place for a dictionary of your notations. Start by writing

```
\newcommand{\newVar}[2]{\newcommand{#1}{\ensuremath{#2}\xspace}}
```

Note how `\ensuremath` was used to make it possible to use these macros both in text mode and in math mode. The `\xspace` at the end of the definition prevents the macro from removing the spaces that follow it.

The next step is to define a small dictionary of notations.

```
\newVar{\groups}{G}
\newVar{\length{\ell}
\newVar{\set{S}
```

14.3 Blackboard Bold font

The blackboard bold font is used for denoting famous algebraic structures: the semi-ring of natural numbers is denoted $\mathbb{N}$, the ring of integers is denoted $\mathbb{Z}$, while the fields of rational, real, and complex numbers are denoted $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$ (respectively). It is simplistic to assume however that

$$\mathbb{C} = \mathbb{R} \times \mathbb{R}.$$

The above text was printed by making the following definitions (the best place for these is in the preamble):

```
\newVar\Naturals{\mathbb{N}}
\newVar\Integers{\mathbb{Z}}
\newVar\Rationals{\mathbb{Q}}
\newVar\Reals{\mathbb{R}}
\newVar\Complex{\mathbb{C}}
```

Note that it was necessary to switch to math mode using with dollar symbols in using the above macros:

```
The blackboard bold font is used for denoting famous algebraic structures:
    the semi-ring of natural numbers is denoted \Naturals, the ring of
    integers is denoted \Integers, while the fields of rational, real,
    and complex numbers are denoted \Rationals, \Reals, and
    \Complex (respectively).
It is simplistic to assume however that \[
    \Complex = \Reals \times \Reals.
\]
```

14.4 New Operators

LaTeX 2ε has many pre-defined mathematical functions, also called operators, including $\sin$, $\lg$, $\min$, etc., which are typeset in ordinary roman font, as in

$$\log_{10} x = \frac{\ln x}{\ln 10}.$$

```
\[ \log_{10} x = \frac{\ln x}{\ln 10}. \]
```

Do declare your operators, use `\DeclareMathOperator`, as in

```
\DeclareMathOperator{\parents}{parents}
```

Take note that this command can be used only in the preamble, and is usually defined after

```
\usepackage{amsopn}
```

Now one can write:

$$\forall x \in M \bullet \operatorname{parents}(x) = 2. \tag{14.1}$$

14.5 Mathematics Checklist

1. Every symbol you use was properly defined and and that it is properly typed.
2. The scope of each definition is clear.
3. Temporary symbols are the only ones with overloaded definitions.
4. No sentence begins with a formula.
5. All inlined formulae are preceded by a nonbreakable space.
6. All displayed formulae are properly punctuated.
7. No displayed formula makes false statements as required for by contradiction.

14.6 Fractions, Binomial Coefficients and Cases

Fractions are generated using the `\frac` command. Do not use the obsolete `\over` command:

$$\frac{\pi^2}{6} = \sum_{n=1}^{\infty} \frac{1}{n^2} \tag{14.2}$$

The $\mathcal{AMS}$ packages make it very easy to generate binomial coefficients using the `\binom` macro. (The obsolete `\choose` macros was a pain.)

$$2^n = \sum_{n=0}^{m} \binom{n}{m} \tag{14.3}$$

Cases constructs are common in definitions. Here is an example how they can be done using the `cases` environment which is provided courtesy of the AMS.

$$\operatorname{level}(x) \equiv \begin{cases} 0 & \text{if } x \in \operatorname{roots}(T) \\ \operatorname{level}(p(x)) + 1 & \text{otherwise} \end{cases} \tag{14.4}$$

14.7 Parenthetical Mathematical Expressions

I like to use the following macros in my preamble to simplify the balancing of parenthetical mathematical expressions.

```
\newcommand\norm[1]{\ensuremath{\lVert#1\rVert}}
\newcommand\abs[1]{\ensuremath{\lvert#1\rvert}}
\newcommand\ceil[1]{\ensuremath{\lceil#1\rceil}}
\newcommand\floor[1]{\ensuremath{\lfloor#1\rfloor}}
\newcommand\set[1]{\ensuremath{\{#1\}}}
\newcommand\angular[1]{\ensuremath{\langle#1\rangle}}
\newcommand\paren[1]{\ensuremath{(#1)}}
```

Here are two simple simple examples showing how these macros might be used:

$$\begin{aligned}\|A\| &= |A|\lceil\log_2|\Sigma|\rceil \\ \|N\| &= \sum_{i=|\Sigma|}^{|\Sigma|+|N|-1} \lceil\log_2 k\rceil\end{aligned} \tag{14.5}$$

norms

```
\begin{equation}
\label{norms}
\begin{split}
    \norm A & = \abs A \ceil{\log_2 \abs\Sigma} \\
    \norm N & = \sum_{i =\abs \Sigma}^{\abs\Sigma + \abs{N} - 1} \ceil{\log_2 k} \\
\end{split}
\end{equation}
```

The `\paren` macro is provided for the sake of completeness.

The capitalized version of these macros resizes the delimiters to match the size of the delimited formula. The lower case version works best in most cases, and should be tried first.

$$G = \langle V, E\rangle = \left\langle \left\{ n^2, \ldots, 2^{n^2} \right\}, \left\{ \langle i, j\rangle | i = j^2 \right\} \right\rangle \tag{14.6}$$

```
\begin{equation}
G = \angular{V,E} = \Angular{
                \Set{n^2, \ldots, 2^{n^2}},
                \Set{\angular{i,j} | i = j^2}
          }
\end{equation}
```

There are at least three different common uses of the vertical bar in mathematical forumulae: for denoting an absolute value, for denoting the norm (double vertical bar) and for conditions in sets. The above macros distinguish between the first two. Here is a useful macro that places appropriate space before and after the vertical bar commonly used in set definitions.

```
\newcommand\st{\;|\;}
```

This macro can then be used as follows

$$\begin{aligned} r(T) &\equiv \{x \in T \mid \nexists y \in T \bullet x \prec_d y\} \\ \ell(T) &\equiv \{x \in T \mid \nexists y \in T \bullet y \prec_d x\} \end{aligned} \tag{14.7}$$

Equa

by typing in the following (which uses the `equation` package):

```
\beqs{hierarchy}
r(T) & \equiv \set{ x \in T \st \nexists y \in T \bullet x \prec_d y}\\
\ell(T) & \equiv \set{ x \in T \st \nexists y \in T \bullet y \prec_d x}\\
\eeq
```

14.8 Equation Numbering

$$1^2 = 1 \tag{14.8}$$ Equa

$$2^2 = 1 + 3 \tag{14.9}$$ Equa

$$3^2 = 1 + 3 + 5 \tag{14.10}$$ Equa

$$4^2 = 1 + 3 + 5 + 7 \tag{14.11}$$ Equa

15 Theorems, Definitions, and Related Creatures

15.1 Defining Environments and Such

Theorems and definitions should be easy for the reader to locate in a large body of text. A common technique is to enumerate each of these individually, and use typography to highlight the theorems.

The `\newtheorem` command defines environments for theorems, lemmas, corollaries (logical conclusions), facts , axioms, definitions, and similarly enumerated creatures. It is common to place all these environment definitions at preamble. These environments can later be used in the document body.

The `\newtheorem`command can be used in one of two ways:

`\newtheorem`{*env_name*}{*Theorem*}[*within*]

`\newtheorem`{*env_name*}[*numbered_like*]{*caption*}

where *env_name* defines an environment name which we can later use to instantiate this environment type. *caption* is the caption for the environment which will appear before the enumeration. *within* allows the binding of this environment's enumeration to a predefined counter instead of the global document counter. *numbered_like* makes the environment use the same counter from another defined theorem environment. For example, many people like to number theorems and lemmas together.

Here is a definition of an environment for axioms, numbered within sections.

```
\newtheorem{axiom}{Axiom}[section]
```

The environment can then be used as follows:

```
\begin{axiom}
There exists a least element in
     every nonempty set of nonnegative integers.
\end{axiom}
```

Axiom 15.1. *There exists a least element in every nonempty set of nonnegative integers.*

The following definition

```
\newtheorem{thm}{Theorem}[]
```

defines a new environment named `thm`, with global numbering which uses the caption "Theorem". To use this definition you may write something like:

```
\begin{thm}[Fermat's Last Theorem]
\label{Theorem:Fermat_Last}
    The equation~$x^{n}+y^{n}=z^{n}$ has no non-zero integer solutions
        for~$x$,~$y$ and~$z$ when $n > 2$.
\end{thm}
```

(Notice how we used the `\label` command to record the theorem number. Also, notice how `~` is used to prevent line breaks before quoted mathematical definitions.)

The output of the above commands would be then:

_Last

Theorem 1 (Fermat's Last Theorem). *The equation $x^n + y^n = z^n$ has no non-zero integer solutions for x, y and z when $n > 2$.*

The optional argument (in square brackets) is used to attribute the mathematical statement to its inventor, to provide the year, theorem name, or any other attribution or comment.

Here is another example:

Theorem 2 (Dirichlet). *Let $f(x)$ be a function define in $[\Leftrightarrow\pi, \pi]$ and let*

$$S_m(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n \cos nx + b_n \sin nx$$

be the partial sums of Fourier' series, where the Fourier' coefficients are $\{a_n\}_{n=0}^{\infty}$ and $\{b_n\}_{n=1}^{\infty}$. Then, the Fourier' series of $f(x)$ at each point x converges to

$$\lim_{m\to\infty} S_m(x) = \frac{f(x^+) + f(x^-)}{2}.$$

```
\begin{thm}[Dirichlet]
 Let $f(x)$ be a function define in~$[-\pi,\,\pi]$ and let \[
    S_m(x) =  \frac{a_0}{2} + \sum_{n=1}^\infty(a_n\cos{nx}+b_n\sin{nx}
 \] be  the partial sums of Fourier' series,
    where the Fourier' coefficients are~$\{a_n\}_{n=0}^\infty$
    and~$\{b_n\}_{n=1}^\infty$.
 Then, the Fourier' series of~$f(x)$ at each point~$x$ converges
    to \[
        \lim_{m\rightarrow\infty}{S_m(x)} = \frac{f(x^+)+f(x^-)}{2}.
    \]
\end{thm}
```

15.2 Referencing Existing Theorems

Because it is usually desired in the text to reference a theorem whose definition has been given, it is common practice to associate a label with that theorem, using the `\label` command.

Let us add a theorem to *Fermat's Last Theorem* as follows:

```
\newtheorem{thm}{Theorem}[]
\begin{thm}[Fermat's Last Theorem]
\label{Theorem:Fermat_Last} The equation $x^{n}+y^{n}=z^{n}$ has
no non-zero integer solutions for x, y and z when $n > 2$.
\end{thm}
```

We will now reference this. For example, the code:

```
The proof of~\ref{Theorem:Fermat_Last} has remained a mystery for
a long time.
```

Will appear as follows:

The proof to Theorem Theorem:Fermat_Last has remained a mystery for a long time.

.

15.3 Exercise in making definitions

LCOM **Definition 15.1 (LCOM).** *Let $v_1, \ldots, v_n$ be the instance variables of class C, and let M_i be the set of methods of C that use v_i. Then,* $\mathrm{LCOM}(c)$ *is the number of unique sets among $M_1, \ldots, M_n$, i.e.,*

$$\mathrm{LCOM}(c) = |\{M_1, \ldots, M_n\}|.$$

We say that v_i is in the same equivalence class as v_j if $M_i = M_j$.

Then, the lack of cohesion in methods *of class C, denoted LCOM(c), is the number of equivalence classes among formed by the intersection of $I_1, \ldots, I_n$. The sets $I_1, \ldots, I_n$ partition the instance variables of C into equivalence classes, where instance variables v_1 and v_2 are in the same equivalence class if*

$$\forall j \in [1, n] \bullet (v_1 \in I_j \Leftrightarrow v_2 \in I_j).$$

Then, the lack of cohesion in methods *of class C, denoted LCOM(c), is the number of disjoint sets formed by the intersection of $I_1, \ldots, I_n$.*

Let $\{I_i\}$ be Let I_i be the set of instance variables used by M_i. Method M_i is not empty, because ...

16 Algorithms

The best package for writing algorithms is `algorithmic`. It is invoked in the preamble by:

```
\usepackage[noend]{algorithmic}
\usepackage[plain]{algorithm}
```

The `noend` option eliminates the **end if**, **end for** etc. The `algorithm` package defines the float `algorithm` environment, which behaves much like figures and tables. The `plain` option makes the layout of the caption of the algorithm look like that of a figure, as in Algorithm Algorithm:PQ, which was produced by the following input:

```
\begin{algorithm}
\caption{PQ-encoding of a hierarchy $T$}
\label{Algorithm:PQ}
\begin{algorithmic}[1]
\STATE {$\groups[1]\leftarrow {\cal P}_u$}
   \COMMENT{An array of the groups (PQ-trees) created so far.}
\STATE {$\lastGroup \leftarrow 1$}
   \COMMENT{The index of the first unused PQ-tree in \groups.}
\FORALL[Find a PQ-tree consistent with type $x$.] {$x \in T$}
\FOR {$g = 1,\ldots, \lastGroup$ }
        \STATE \textsf{reduce}($\groups[g]$,$\descendants(x)$)
        \STATE \textbf{exit loop if} \textsf{reduce} succeeded
    \ENDFOR
    \STATE {$g_x \leftarrow g$}
    \IF [Start a new universal PQ-tree]
    {g = \lastGroup}
        \STATE
            $\lastGroup \leftarrow \lastGroup + 1$;
            $\groups[\lastGroup] \leftarrow {\cal P}_u$
        \ENDIF
\ENDFOR
\FOR[{Assign a unique id to each type in each group}]
    {$g = 1, \ldots, \lastGroup-1$}
    \STATE{$ \currentId \leftarrow 1$}
        \COMMENT{The first unused id in the group $\groups[g]$.}
    \FORALL {$x \in \frontier(\groups[g])$}
        \STATE $\id_x[g] \leftarrow \currentId$;
                $\currentId \leftarrow \currentId+1$
    \ENDFOR
\ENDFOR
\FORALL[Assign an interval to each type $x$ ]{$x \in T$ }
    \STATE $l_x\leftarrow \min\set{\id_y[ g_x ] \st y \in \descendants(x)}$
    \STATE $r_x \leftarrow \max\set{\id_y[ g_x ] \st y \in \descendants(x)}$
\ENDFOR
\end{algorithmic}
\end{algorithm}
```

Many customization of the algorithmic package are possible. The ones used in producing Algorithm 1 were:

```
\renewcommand\algorithmiccomment[1]{// \textit{#1}}
\renewcommand\algorithmicif{\textbf{If}}
\renewcommand\algorithmicfor{\textbf{For}}
\renewcommand\algorithmicforall{\textbf{For all}}
```

Resist the temptation of applying idiosyncratic programming constructs and idioms, such as `++`, `:=`, etc. in writing algorithms. The purpose of writing algorithms is to explain the algorithm, not the implementation. It is not only permissible but also mandated to use the power of mathematical formulae, and to abstract away from details of the implementation. Instead of writing

s **:=** 0
For $i \leftarrow 1$ to n **do**
  s **+:=** $a[i]$

write

$s \leftarrow \sum_{i=1}^{n} a[i]$.

More generally, mathematical conventions are preferred over programming conventions. Do not choose "meaningful", mutliple-character names to variables used in algorithms.

`Algorithm:PQ`

1: $\mathbf{G}[1] \leftarrow \mathcal{P}_u$ // *An array of the groups (PQ-trees) created so far.*
2: $\ell \leftarrow 1$ // *The index of the first unused PQ-tree in* $\mathbf{G}$.
3: **For all** $x \in T$ **do** // *Find a PQ-tree consistent with type* x.
4: **For** $g = 1, \ldots, \ell$ **do**
5: reduce($\mathbf{G}[g]$,$\mathrm{descendants}(x)$)
6: **exit loop if** reduce succeeded
7: $g_x \leftarrow g$
8: **If** g = ℓ **then** // *Start a new universal PQ-tree*
9: $\ell \leftarrow \ell + 1$; $\mathbf{G}[\ell] \leftarrow \mathcal{P}_u$
10: **For** $g = 1, \ldots, \ell \Leftrightarrow 1$ **do** // *Assign a unique id to each type in each group*
11: $\mathrm{id} \leftarrow 1$ // *The first unused id in the group* $\mathbf{G}[g]$.
12: **For all** $x \in \mathrm{frontier}(\mathbf{G}[g])$ **do**
13: $\mathrm{id}_x[g] \leftarrow \mathrm{id}$; $\mathrm{id} \leftarrow \mathrm{id} + 1$
14: **For all** $x \in T$ **do** // *Assign an interval to each type* x
15: $l_x \leftarrow \min\{\mathrm{id}_y[g_x] \mid y \in \mathrm{descendants}(x)\}$
16: $r_x \leftarrow \max\{\mathrm{id}_y[g_x] \mid y \in \mathrm{descendants}(x)\}$

Algorithm 1: PQ-encoding of a hierarchy T

17 Programs

Actual program code is included in paper is concerned with the unique features of a specific programming language.

The `prog2tex` package, available as `/home/yogi/TeX/prog2tex.sty`, is very handy in typing C and C++ programs. A famous beginner's C program is shown in Fig. 17.1. Note how all the keywords are printed in boldface, text in strings is in italics and that comments are typeset in times new roman font.

`Figure:famous`

With great effort, one can manually generate this beautiful output. The `prog2tex` package allows you to simply type in program code, between `\CPP` and `\END` macros, as in Fig. 17.2. There is no need to worry about the special meaning of characters such as #, <, {, etc. to TEX.

`Figure:input`

Then, you will need to run the program `prog2tex`, available as `/home/yogi/bin/prog2ex`, on your input file:

```
% prog2tex howto.tex
```

This will create an auxiliary file named `howto.prg`, with a macro definition generating the above pretty print. In addition, running `prog2tex` will also *modify* `howto.tex`, adding after the `\END` command in your input (Fig. 17.2) a call to a macro with a funny name such as `\PROGxba`. The little secret that `prog2tex.sty` applies is that it simply instructs LATEX 2_ε to completely ignore anything starting at `\CPP`, and ending at `\END`. The `\PROGxba` macro, defined in `howto.prg` does the whole work of generating the actual layout. The `prog2tex` package just makes sure that the `howto.prg` file with all its macros is input before LATEX 2_ε begins processing the main file.

`Figure:input`

18 The elements of style

18.1 Elementary Rules of Usage

1. Form the possessive singular of nouns by adding 's.
2. In a series of three or more terms with a single conjunction, use a comma after each term except the last.
3. Enclose parenthetic expressions between commas.
4. Place a comma before and or but introducing an independent clause.
5. Do not join independent clauses by a comma
6. Do not break sentences in two
7. A participial phrase at the beginning of a sentence must refer to the grammatical subject
8. Divide words at line-ends, in accordance with their formation and pronunciation

18.2 Elementary Principles of Composition

1. Make the paragraph the unit of composition: one paragraph to each topic
2. As a rule, begin each paragraph with a topic sentence; end it in conformity with the beginning
3. Use the active voice

```
#include <stdio.h>/*** A small C program to print its argumnts      */main(char *argv[], int argc) {
    int i;
    for (i = 0; i < argc; i++)          (void) printf("Arg %3d: %s\n", i, argv[i])
    return 0; }
```

Figure 17.1: A C program to print the command line arguments

Figure

```
\CPP
#include <stdio.h>
/*
** A small C program to print its argumnts
*/
main(char *argv[], int argc)
{
    int i;

    for (i = 0; i < argc; i++)
        (void) printf("Arg %3d: %s\n", i, argv[i])

    return 0;
}
\END
```

Figure:famous

Figure 17.2: Input for creating the Fig. 17.1

Figure

4. Put statements in positive form
5. Omit needless words
6. Avoid a succession of loose sentences
7. Express co-ordinate ideas in similar form
8. Keep related words together
9. In summaries, keep to one tense
10. Place the emphatic words of a sentence at the end

19 Conclusions

Acknowledgments I am grateful to Sally Tadmor, Gil Gattegno and Uriel Cohen for valuable contributions to this document.

References

Grif [1] D. F. Griffiths and D. J. Higham. *Learning LaTeX*. SIAM, Philadelphia, 1997.

Lamp [2] L. Lamport. *LaTeX: a document preparation system*. Addison-Wesley Publishing Company, 2nd edition, 1994.

Oet [3] T. Oetiker, H. Partl, I. Hyna, and E. Schlegl. *The Not So Short Introduction to LaTeX 2ε*. Free Software Foundation, 1998.

A First appendix

B Second appendix

Contents

1 Introduction **1**

2 Basics **3**
2.1 How to run LaTeX 2ε . . . 3
2.2 Check list . . . 4

3 Plain Text **6**
3.1 Punctuation . . . 6
3.2 Quoting . . . 7
3.3 Dashes . . . 7

4 Special Characters **8**
4.1 Comments . . . 9
4.2 Commands and Environments . . . 9
4.2.1 The `document` Environment . . . 10
4.2.2 The `abstract` Environment . . . 10
4.2.3 Environments as Scope Delimiters . . . 10
4.2.4 Making your own commands . . . 10
4.3 Non-breakable Space . . . 12

5 Organizing Your Article **13**
5.1 Sectioning . . . 13
5.2 The Components of a Paper . . . 13
5.3 Lists . . . 15

6 Installing New Packages **16**
6.1 The `labels` Package . . . 17

7 The sentence structure **18**
7.1 The Meat and the Bones . . . 18
7.2 Tips . . . 20

8 Discussion of a Home Assignment of a Student **22**
8.1 Title . . . 22
8.2 General comments . . . 22
8.3 Examples . . . 23

9 Rephrasing a Sentence–Examples **24**

10 Discussion of a Home Assignment of a Student **27**
10.1 Title . . . 28
10.2 General comments . . . 28

11 Tables and Figures **28**
11.1 Tables . . . 29
11.2 Figures . . . 32

12 Advanced topics in LaTeX **35**
12.1 Numbering . . . 35
12.2 LaTeX's auxiliary files . . . 36
12.3 Building a Bibliography . . . 36
12.4 Creating a Table of Contents . . . 37

13 Mathematical Formulae **37**
13.1 Subscripts and Superscripts . . . 37
13.2 Mathematical Symbols . . . 38
13.3 Typefaces for Mathematical Formulae . . . 38

14 Phrasing Definitions and Theorems **38**
14.1 Escaping the Mathematical Special Characters . . . 38
14.2 The Programming Metaphor for Mathematical Notation . . . 39
14.3 Blackboard Bold font . . . 40
14.4 New Operators . . . 41
14.5 Mathematics Checklist . . . 41
14.6 Fractions, Binomial Coefficients and Cases . . . 41
14.7 Parenthetical Mathematical Expressions . . . 42
14.8 Equation Numbering . . . 43

15 Theorems, Definitions, and Related Creatures **43**
15.1 Defining Environments and Such . . . 43
15.2 Referencing Existing Theorems . . . 44
15.3 Exercise in making definitions . . . 45

16 Algorithms **45**

17 Programs **48**

18 The elements of style **48**
18.1 Elementary Rules of Usage . . . 48
18.2 Elementary Principles of Composition . . . 48

19 Conclusions **50**

A first **52**

B second **52**

List of Tables

3.1 Formatting characters 6
4.1 Special characters 9
11.1 The usage frequency of some English words. 31

List of Figures

5.1 The schematic structure of a LaTeX 2_ε article 14
6.1 Boats used to cross the Atlantic. 19
11.1 The mid-term exam results of the course ”Set Theory” for the previous (left column) and the current (right column) semesters. 33
11.2 The clown. 34
11.3 Plot of function. 34
17.1 A C program to print the command line arguments 49
17.2 Input for creating the Fig. 17.1 49

Figure:famous

List of Algorithms

1 PQ-encoding of a hierarchy T 47