LaTeX- Does Your Documents Good

Orr Dunkelman

orrd@vipe.technion.ac.il

Introduction

In the beginning D. Knuth created TEX; And the TEX was hard to use, and people had hard time writing documents in TEX; And L. Lamport said, Let there be LATEX: and there was LATEX;

And Lamport saw the LATEX, that it is good: and Lamport added new features and finally wrote new version; And Lamport called the latest version he wrote LATEX2.09, and the evening and the morning - chapter 1.

And the people saw they want more than what Lamport has given them; And they gathered to write LATEX3, but god himself prohibited that. and all they succeeded was LATEX2$_\epsilon$

Introduction (cont')

TEX was first written by D. Knuth in 1977 TEX is a type setting system, which was needed by Knuth to finish his "The Art of Computer Programming" 6-volume book.

TEX can be best defined as a programming language for documents. In order to shape your document the way you want it, you need to use special commands. However, many needed extensions to TEX, and among the known extension one can find LATEX and AMS-TEX (which was later evolved into AMS-LATEX).

Introduction (cont')

Currently most of the academic world uses LaTeX (2.09 or 2_ϵ) as the document preparation utility.

Advantages of LaTeX:

- Unix program which needs almost no CPU power.
- Has a vast number of utilities which allows creating Postscript and PDF files.
- Large number of easy to access mathematic symbols.
- Flexibility - any mathematical/scientific text can be processed using LaTeX.

LaTeXhas also graphical interface called LyX.

Getting Started

Use your favorite editor (vi, emacs, ...) to create try.tex.

write down the following text in the file:

```
\documentclass[]{report}
\begin{document}
Hello World
\end{document}
```

Save it, and run *latex file.tex*. A dvi file containing the output of LaTeX will be created. You can see it using *xdvi*. You can take the .dvi and make it a postscript using *dvips*.

Getting Started (cont')

What have we done here?

The first line announces the type of the document being processed - \documentclass[]{}.

In {} we put the type of the document - it can be one of the following article, book, letter, report or seminar (slides like those you are reading now). The difference between report and article is the length (articles are shorter).

In [] we put the additional packages we want to use, for example [a4] means we want the output to be in the form of a4 pages, [psfig] asks LaTeX to allow us including postscripts into our document.

Getting Started (cont')

As you might have noticed, each command starts with \, This is a reserved symbol, and the only way to integrate \ into the text is using ad-hoc solutions (like the one I'm using now and you'll two once we finish this lecture . . . I hope).

The first command after you announce what document type you wish for, is the *begin*{*document*}. This structure of command is very common to LaTeX, when we begin area of special interest to LaTeX (like the text to be processed) we announce it's beginning using the *begin*{*What ever begins here*}). Logically enough, the end of the area is marked by *end*{*Same as the begin command*}.

Getting Started (cont')

The \begin{document} (and the corresponding \end command) announce the beginning (end) of the document. Till this command the headers of LaTeX are put (announcing new commands, packages to be used, document style, etc.) and now the real document starts.

Writing a Document

Assume for a moment that the right packets have been loaded (For novices the announcement of *documentclass* is more than enough). We want to be able to write a document.

After the *begin*{*document*} command, we can write any text we wish. The text can be plain English (letter to the government), pure mathematics (Solving Fermat's last theorem), has graphical content (article about analyzing people's faces) or a good book (Like the "LaTeX User's Guide and Reference Manual" by L. Lamport).

We will not discuss books in this lecture, and divided the world between the three main groups - articles (and reports), slides (seminar) and letters (if time permits).

Articles - Sections

LaTeX was created by Lamport in order to allow people to write technical reports and scientific papers using the TeX system. Hence, LaTeX supports (in the report and article class) in dividing the document into sections, subsections, subsubsection, and appendixes.

To define a new section one just declare *section*{*Title of Section*}. New subsection uses the command *subsection*{*Title of Subsection*}. Subsubsection is created in a very similar way.

LaTeX keeps track of the sections numbering automatically. It allows you (using a referring mechanism presented later) to move sections in your document without need to change anything. Another feature is that the number of the section is presented near the section title. To avoid this add * between the section (or subsection command) and the {}.

Article - Referring

The referring mechanism resembles VI's method. We leave a mark in the place we want to refer to, and then we point to the mark.

Marking is done using the command *label*{*Label Code*}, and referring is done using the command *ref*{*Label Code*}.

LaTeX keeps separate Label codes for sections, tables and figures. Meaning that a label inside a table scope will get the number of the table among the tables.

Article - Basic Functions

We just write a text. However, there are characters you cannot type in as they are, due to the fact that they have different meaning in LaTeX.

The symbols are $ & % # _ { } ~ ^ \.

The first 7 ($ ... }) can be written be writing in the text file - \$ ...\}. The other three requires several tricks in order to write them, including entering to mathematical mode.

Article - Mathematical Mode

We can enter mathematical mode either by using $ *math text* $, or by using \[command (ending it with \]).

$ doesn't open new line and remain in the same line, just like in this case $x + y = 2^5$. However, \[just starts a new line, and the formula is centerlized just like in this case

$$\alpha + \beta^{\gamma_1} = 45^\circ \cdot f^2(n^{2^{2^2}})$$

An equivalent to $ is the \(, \) pair.

Article - Mathematical Mode (cont')

In the mathematical mode we can put numbers, operators, Greek letters (like α which is created using the command \alpha), special mathematical symbols (like integral, \int) and sub/super-scripts (like 2^5, which is done by ^{5}).

In the previous example you have witnessed the ability to have super super scripts, and this is done by putting in the ^{} segment additional ^{}. Sub script is done similarly with the character _ (replaces ^).

Comment: In mathematical mode there are no space, and the text is written in italic. In order to space use one of the commands *quad or* ~. In order to have a text a normal spacing, put the text in \mbox{Text Here}. Thus you can get:

$\text{Example}^{\text{To a very}}_{\text{Clever Way}} \; To$ eat apples

Article - Scopes

Scope can be defined using two techniques. The first is to get into mathematical mode. In math mode, everything you do remains inside the math mode (you can't have super script of sub script if you're not in the math mode).

Another way is to use {,}. This enables changing of font/size for a scope, for example **{\it abc }** would produce *abc*.

You can of course use the \begin \end method, by using **\begin{it} abc \end{it}** to achieve the same text.

The codes are as follows:

Code	Affect	Code	Affect
\rm	Roman	\em	*Emphasize*
\bf	**Bold**	\it	*Italic*
\sl	*Slanted*	\sf	Sans Serif
\sc	SMALL CAPS	\tt	`Type Writer`
\tiny	Tiny	\scriptsize	Script Size
\footnotesize	Footnote Size	\small	Small
\normalsize	Normal Size	\large	large
\Large	Large	\LARGE	LARGE
\huge	huge	\Huge	Huge

Table 1: Codes of Text Manipulation in LaTeX

Article - Tables

The table itself is called tabular (command *\$\$\\begin{tabular}* and the corresponding *\\end{tabular}\$\$*). However, we can call a block a table (even if it has no tabular in it!) where we design lines which will be related to the tabular.

To create the following table

Lines before

first field	second field	third field
Both fields		Third Field
		Banana
Bo		

Table 2: Example

Article - Tables (cont')

The syntax would be:

```
\begin{table}
Lines before
$$\begin{tabular}{lcr}
\hline
first field & second field & third field\\
\hline
\multicolumn{2}{c}{Both fields}&Third Field\\
    &    & Banana\\
\cline{1-2}
Bo\\
\hline
\end{tabular}$$
\caption{Example}
```

```
\label{tab:example}
\end{table}
```

Article - Tables (cont')

And we can address the table using *ref*{*tab:example*}.

We can also have vertical lines between columns if we define in the tabular section {|l|||c||r}, which will give one vertical line before the first field, three between the first and the second, and two between the second and the third.

As you can see, change of column is done by the & character. End of line is done using \\\\. Note that you don't need to have all the columns to be filled in order to end line.

\\hline draws an horizental line.

\\multicolumn{how many columns}{l or c or r} unite columns, while \\cline{which column to start-which column to end}, tells LaTeX to put there horizental line (but on several columns).

Article - More on Equations

Equations can also be refered (with their own counters). The beginning command is $\backslash begin\{eqnarray\}$ (without counting - put * between the begin and the {}).

Eqnarray opens an array with 3 columns (an array is a table in the equation/math mode of LaTeX), again changing column is done using &, and in the end of line \\

Using an array the following can be done:

$$x = \begin{cases} y & \text{if } y > 0 \\ z \cdot y & \text{otherwise} \end{cases}$$

More on Equations (cont')

The above was generated from the following LaTeX sequence:

```
\[
x= \left\{ \begin{array}{ll}
 y & \mbox{if $y>0$}\\
 z \cdot y & \mbox{otherwise}\\
\end{array}
\right.
\]
```

Note that the *left* can come with (,[, or | . It needs to be closed be appropriate *right* (blank which is defined as . or with some),],}).

Comment : it is allowed to have an array inside an array (inside an array, . . .).

Bibliography

As bibliography is quite tricky, LaTeX keeps a special tables for bibliography entries. The refering is done using the command *cite*{*Cite Key*}. The “Cite Key” can be a number (1) or the initials of the authors (KP83).

Each \cite need to have a corrseponding entry in the \begin{bibliography} section (ends with the appropiate \end) where each item begins with \bibcite{Cite Key}.

Enumrators and List of Items

1. It is very common to discuss algorithm as steps.
2. As you might think of, LaTeX give you tool for doing this.
3. The \begin command is $\backslash begin\{enumerate\}$.
4. Each new item is created using $\backslash item$.
5. If you want just list without numbers - use $\backslash begin\{itemize\}$.
6. Don't forget the right \end command.

Article - Paragraphs and Page Format

LaTeX ignores double spaces in the source .tex file. It assumes that when you want double space you will use the right technique ($\sim\sim$ for example).

In order to break line - \\.

A new paragraph starts when you have an empty line between the text. If you want LaTeX to avoid indenting it, you can use the command $\backslash noindent$.

New Commands

One can add commands (macros) to LaTeX or rewrite commands using the command (before the document starts) - *newcommand*{*new command name*}{*What it does*}. Note that the command can stand for just putting a text. For example, the for example can be put as a macro:

```
\newcommand{\fe}{For Example}
```

And once you use \fe it will be replaced with "For Example".

To redefine a predefined command we use *renewcommand.*

Note that it is most wise to use \mbox in defining macros (and new commands containing text) due to LaTeX properties.

New Commands (cont')

One can also have newcommand with parameters:

```
\newcommand{\exp}[2]{\mbox{$#1^{#2}}}
```

Will cuase \exp{2}{3} to produce 2^3.

References

[1] *LaTeX User's Guide & Refernce Manual*, L. Lamport, Addison Wesley, 1986.

[2] *The Not So Short Introduction to LaTeX 2ε*, T. Oetiker et al., *http://ctan.tug.org/tex-archive/info/lshort/english/lshort.pdf*