Math 290: LaTeX Seminar Week 8

Theorem-Like Environments and Defining New Commands

Justin James

March 1st, 2011

Minnesota State University Moorhead

Theorem-Like Environments

- LaTeX has several pre-defined *environments* that allow us to quickly typeset theorems or theorem-like structures, without having to manually set the typeface and numbering ourselves.

- Environments start with the command:
  \begin{environment name}
  and end with the command:
  \end{environment name}.

Theorem-Like Environments

- One such command in LaTeX is:

  ```
  \newtheorem{Theorem name}{theorem title}
  ```

- In this command, ‘Theorem name’ is the name used to call the environment.

- ‘Theorem title’ is the title or name that is actually printed (along with with a “counter”) when the document is compiled.

An Example:

- Open a new document containing your standard preamble.

- In the body of your document, type:

  ```
  \newtheorem{thm}{Theorem}
  ```

  into the body of your document.

- Then, type in a similar command to define the “axiom” environment:

  ```
  \newtheorem{ax}{Axiom}
  ```

An Example:

- Next, type in the following and then build:

  ```
  \begin{thm}[The Fundamental
  Theorem of Calculus]
  $$\int_{a}^{b} f(x)\, dx = F(b)-F(a)$$
  \end{thm}
  ```

- Now type in commands and build to produce the following:

  Axiom 1. *All fishes, except sharks, are kind to children.*

  Axiom 2. *Kangaroos are not suitable for pets.*

Theorems, continued

- Notice that LaTeX keeps track of the numbering for you.

- When you add or remove theorems, the numbering throughout the document is automatically updated.

- We will eventually learn a way to refer to theorem numbers.

- When used, reference labels will also be automatically updated.

Theorems, continued

- There is an optional argument that gives theorems titles of the form **Theorem 4.2** (here the 4 refers to some outside counter, like a chapter, and the 2 means that it is the second numbered theorem within that chapter.

- The syntax is:
  `\newtheorem{thm2}{Theorem}[enumi]`
  where enumi is the counter being used
  (in this case, the first level of an enumerate environment).

- Define a new environment “thm2” using the syntax above.

- Then try combining this environment with an enumeration to create the following:

  1. This is the first enumerated item.

  2. This is the second enumerated item.

  3. This is the third enumerated item.

     Theorem 3.1. *This is the first numbered theorem after item three.*

     Theorem 3.2. *This is the second numbered theorem after item three.*

  4.

     Theorem 4.1. *This is yet another theorem statement - How did it get numbered?.*

- Note that we referred to a specific enumeration level in our environment definition.

- Other counters can be used, like sections and chapters for larger documents.

Theorems, continued

- If you don't want to take the time to define environments yourself, you can use the `amsthm` package.
  - Along with the standard environments, this package defines a `newtheorem*` version, used for unnumbered theorems
  - It also defines three environment styles:
    - plain (bold title, then italics in the body)
    - definition (bold title, then normal text in the body)
    - remark (italicized title, then normal text in the body)
  - A user can still manually define other theorem styles.

- - The “amsthm” package also defines the `swapnumbers` command, which puts the numbers *before* the theorem (as in: **1 Theorem**).
  - Finally, it defines a `proof` environment. This environment:
    - is unnumbered
    - it starts with *Proof*
    - it ends with □.

Manually Defining Your Own Commands:

- The syntax for defining a command is: \newcommand{\name}[#args][opt]{def}

  Note: LaTeX will not allow you to redefine a command that has already been defined internally.

- **Example:** \newcommand{\ds}{\displaystyle}

- When placed in the preamble, the command is globally defined (applies to the entire document).

- When placed within an environment, it is defined only within that environment.

- When placed elsewhere in the body of a document, it can only be used from then on.

New Commands with Arguments

- The “#args” part of the newcommand syntax indicates the number of arguments that are required to be supplied when using the command (each argument should be put within a separate “{}”).

- Each argument will be referred to separately in the definition of the command by using: #1, #2, etc..

- The command `\ensuremath` ensures the command will **always** be carried out in math mode (whether you call the command inside $ signs or not).

Examples of Commands with Arguments

- Type the following into your document:

  ```
  \newcommand{\repdec}[1]
  {\ensuremath{0.\overline{#1}=\frac{#1}{99}}}
  ```

- And now call your new command with: `$\repdec{63}$`

  What happens when you build? Notice that this is a command with a single argument (input).

  Try changing the input value and see what happens to the output.

Examples of Commands with Arguments

- As another example, here is a command requiring 4 inputs:

  ```
  \newcommand{\cfrac}[4]
  {\ensuremath{\frac{\frac{#1}{#2}}
  {\frac{#3}{#4}}}}
  ```

- Add this command definition to your sample document.

- Then, test out your new `\cfrac` command using some different input values. What does this command do?

Example of a command with an optional argument

- The `\newcommand` also allows you to define commands with *optional* arguments (arguments that are available for use but not absolutely required).

- For example, try adding the following to your sample document:

  ```
  \newcommand{\subvec}[3][x]
  {\ensuremath{#1_{#2}, \ldots, #1_{#3}}}
  ```

- The first of the three arguments is optional since a default value has been supplied.

- If a new value for this optional argument is **not** supplied, the default value of x will be used. Otherwise, the new input value will be used.

Example of a command with an optional argument

- Try calling this command three times using the following inputs:

  ```
  \subvec[x]{1}{n}
  ```

  ```
  \subvec[y]{1}{n}
  ```

  ```
  \subvec{1}{n}
  ```

The Renewcommand Command

- There is also the `\renewcommand` command.

- Here is a command that Dr. Hill uses when she runs out of alphabet on her review sheets:

  ```
  \setcounter{enumi}{0}
  \renewcommand{\labelenumi}
  {(\alph{enumi}\alph{enumi})}
  ```

- Try using this command to create the following enumeration:

(a) First

(b) Second

(aa) Third

(bb) Fourth

- **Be careful** when using `renewcommand`. You can use it to accidentally overwrite standard LaTeX commands!

New environments

- Finally, the following syntax can be used to define a new LaTeX environment:

  ```
  \newenvironment{envname}[narg][opt]
  {begdef}{enddef}
  ```

- We can also define and make use of new counters using the command `newcounters`.

- These options are part of what makes LaTeX high customizable and useful.