

The Use of *newcommand* in GELLMU

1 Introduction

In GELLMU the command *newcommand* is a meta-command that is handled by macro-like pre-processing in the GELLMU syntactic translator. This pre-processing takes place very early after the processing of the required meta-command *documenttype* and after the insertion of any (relatively rare) internal SGML declaration subset.

2 First Examples

The markup for the field \mathbf{Q} of rational numbers is a *newcommand* invocation " $\backslash\mathbf{Q}$ " without arguments that expands to " $\backslash\mathit{regch}\{\backslash\mathbf{Q}\}$ ".¹ Newcommand names are case-sensitive. Here the newcommand invocation " $\backslash\mathit{q}$ " expands to " $\backslash\mathit{regch}\{\backslash\mathit{slnt}\{\mathbf{Q}\}\}$ ", which yields the ad-hoc math symbol Q .

Markup to represent the general linear group may be a *newcommand* with definition

```

$$\backslash\mathit{newcommand}\{\backslash\mathit{gl}\}[2][2]\{\backslash\mathit{mbox}\{\mathit{GL}\}_{\#1}(\#2)\} .$$

```

When invoked with " $\backslash\mathit{gl}\{3\}\{F\}$ ", it yields $\mathit{GL}_3(F)$.

Mention of the group $\mathit{GL}_2(F)$ is obtained alternatively here with the minimized markup invocation " $\backslash\mathit{gl}\{F\}$ " relying on the default value of the first argument given in the definition of " $\backslash\mathit{gl}$ ".

3 Some Details

With the *newcommand* definition

```

$$\backslash\mathit{newcommand}\{\backslash\mathit{Q}\}\{\backslash\mathit{regch}\{\backslash\mathbf{Q}\}\}$$

```

the markup $\backslash\mathit{Q}$; may always be used instead of $\backslash\mathbf{Q}$ and must be used if the \mathbf{Q} is followed immediately by characters that could have the effect of making the ‘ \mathbf{Q} ’ in the markup look simply like the first character in a larger string. However, the markup $\backslash\mathit{Q}$;{} is incorrect unless *regch* is a command that takes two arguments. For it expands to

```

$$\backslash\mathit{regch}\{\backslash\mathbf{Q}\}\{\}$$

```

¹The command *regch* is an alternative form of *mbox* in the didactic GELLMU document type definition that differs from the latter in that it is permitted as the content of an “accent” tag as in the example $\overline{\mathbf{Q}}$, which is written in the GELLMU source here with $\backslash\mathit{ovbar}\{\backslash\mathbf{Q}\}$.

3.1 What Markup Can Follow a *newcommand*?

What characters can safely follow the name of a *newcommand*? Of course, a following ‘{’ can only mean the beginning of an argument to the command. On the other hand the ‘{’ can be escaped.

Some examples:

Markup	Yield
$\backslash Q$	Q
$\backslash Q;xy$	Qxy
$\backslash Q;\$xy$	Qxy
$\backslash Q;;$	$Q;$
$\backslash Q;\$$	$Q;$
$\backslash Q\backslash bsl;$	$Q\backslash$
$\backslash Q\backslash\{$	$Q\{$
$\backslash Q\backslash}$	$Q}$
$\backslash Q\backslash lsb;$	$Q[$
$\backslash Q\backslash rsb;$	$Q]$
$\backslash Q!$	$Q!$
$\backslash Q@$	$Q@$
$\backslash Q\#$	$Q\#$
$\backslash Q\%$	$Q\%$
$\backslash Q^5$	Q^5
$\backslash Q_p$	Q_p
$\backslash Q\backslash und;$	$Q_$
$\backslash Q\backslash crt;$	$Q^$
$\backslash Q*$	$Q*$
$\backslash Q-$	$Q-$
$\backslash Q(t)$	$Q(t)$
$\backslash Q+$	$Q+$
$\backslash Q=$	$Q=$
$\backslash Q\$=$	$Q=$
$\backslash Q,$	$Q,$
$\backslash Q.$	$Q.$
$\backslash Q?$	$Q?$

3.2 Comments in *newcommand* definitions

It is useful here to comment on GELLMU, which is a bridge from \LaTeX to SGML, as it compares to \LaTeX . \LaTeX usage of *newcommand* sometimes takes the form

```
\newcommand{\foo}[2]{%
This is an instance of the \emph{newcommand} ‘foo’ %
with first argument ‘#1’ and% Space following a newline is swallowed
second argument ‘#2’.
```

This usage is useful in that it gives the author a way, especially with empty comments, to make the source document more readable.

With GELLMU as a general matter such use of comments in regular command arguments may have side effects if the author is not careful. Because *newcommand* is a meta-command handled by the GELLMU syntactic translator, this use of comments will not have side effects.

Since *newcommand* in GELLMU is a meta-command, it is **not** necessary to restrict an instance of *newcommand* to a document preamble. However, any invocation of a command defined with *newcommand* must be preceded by the definition.

The definition of `\foo` appears in this document just before the display of its definition, and the next sentence is the invocation "`\foo{cat}{dog}`". This is an instance of the *newcommand* "foo" with first argument 'cat' and second argument 'dog'.

Note that the comment "Space following ..." on the second line of the value string absorbs the space at the beginning of the third line with the result that the words *and* and *second* are run together. This phenomenon mimics L^AT_EX; however, such comment "overrun" is the behavior of the GELLMU syntactic translator only inside a *newcommand* value string. In all other contexts there is no comment overrun in GELLMU, and space at the beginning of a line following a comment is significant.

3.3 Newlines in newcommand definitions

It is important to note, however, that *anything* inside the value argument of a *newcommand* definition is propagated by the GELLMU syntactic translator to every invocation. With basic GELLMU newlines inside a newcommand value string are removed since SGML does not acknowledge a newline as a special form of white space and since newlines in a value string, when propagated to all invocations cause extra lines in the GELLMU syntactic translator output, thereby blocking the goal of line number alignment between GELLMU source and the SGML output, which is important for the interpretation of SGML parser error messages.

In regular GELLMU, newlines can be significant. For this reason newlines in newcommand definitions are not removed. However, the use of newlines inside value strings is deprecated since they break line number alignment between the GELLMU source and the output of the GELLMU syntactic translator. There are other ways to achieve the effect of a newline inside a newcommand value string.