Makros in LaTeX

Luca Kiebel

6. Februar 2018

Inhaltsverzeichnis

1 Einleitung

Die meisten LaTeX Befehle sind einfache Worte mit einem Backslash davor:

Listing 1: Befehle

```
In einem \LaTeX \ Dokument gibt es verschiedene \textbf{Befehle}, \\
die Text verschieden \textit{darstellen} können.
```

In einem LaTeX Dokument gibt es verschiedene **Befehle**,
die Text verschieden *darstellen* können.

Makros sind selbstdefinierte Befehle, die entweder alte Kommandos überschreiben oder neue definieren. Sie sind standardmäßig in LaTeX integriert und müssen nicht als Paket eingefügt werden. Es gibt drei wichtige Befehle zum erstellen von Makros:

\newcommand erstellt einen neuen Befehl und gibt einen Fehler aus, wenn der Befehl schon existiert.

\renewcommand erneuert einen Befehl mit der übergebenen Programmsequenz und gibt einen Fehler aus, wenn der Befehl noch nicht existiert.

\providecommand erstellt ebenfalls einen neuen Befehl, wird aber ignoriert, sollte der Befehl schon existieren.

Informationen aus dem folgenden Teil sind entnommen aus **wiki:macros**

2 Befehle definieren

LaTeX kommt mit einer großen Anzahl von Befehlen für viele Aufgaben, dennoch ist es manchmal notwendig, einige spezielle Befehle zu definieren, um wiederholte oder komplexe Formatierungen zu vereinfachen und zu verkürzen. Hier kommen Makros ins Spiel.

2.1 Einfache Makros

Neue Befehle werden durch die `\newcommand`-Anweisung definiert. Will man zum Beispiel alle Reellen Zahlen ausgeben, so kann der Befehl \(\mathbb{R}\) einfach durch das Makro \R ersetzt werden:

Listing 2: newcommand

```
\newcommand{\R}{\(\mathbb{R}\)}
Die Zahl n ist Teil der Menge der reellen Zahlen \R
```

Die Zahl n ist Teil der Menge der reellen Zahlen $\mathbb{R}$

Dieser neue Befehl kann dann im LaTeX Code an jeder Stelle genutzt werden.
Die Anweisung hat 2 Parameter, die den neuen Befehl definieren:

\R ist der Name des neuen Befehls

\(\mathbb{R}\) ist die Abfolge an Befehlen, die das Makro ausführt.

2.2 Makros mit Parametern

Es ist auch möglich, Makros zu definieren, die Parameter annehmen, um die übergebenen Daten in dem Makro zu verarbeitern:
Um einen einfacheren Weg zu haben Mengensymbole in LaTeX auszugeben kann man sich einen speziellen Befehl definieren.

Listing 3: Mengenzeichen

```
1 \newcommand{\mz}[1]{\(\mathbb{#1}\)}
2 Wenn die Zahl n Teil der Menge der ganzen Zahlen \mz{Z} ist, ist sie auch
    Teil der Menge der reellen Zahlen \mz{R}
```

Wenn die Zahl n Teil der Menge der ganzen Zahlen $\mathbb{Z}$ ist, ist sie auch Teil der Menge der reellen Zahlen $\mathbb{R}$

Die Zeile \newcommand{\mz}[1]{\(\mathbb{#1}\)} definiert einen Befehl, der ein Parameter akzeptiert.

\mz ist der Name des neuen Befehls.

[1] ist die Anzahl der akzeptierten Parameter

\(\mathbb{#1}\) ist der ausgeführte Befehl. Anstatt, dass wie zuvor der Buchstabe für das Mengensymbol fest gesetzt ist, wird hier automatisch der erste Parameter eingesetzt.

Ein Makro kann bis zu 9 Parameter haben, auf die dann mit #1, #2 usw. zugegriffen werden kann.

2.3 Makros mit optinalen Parametern

Makros können auch so erstellt werden, dass nur ein Teil der Parameter beim Aufruf gesetzt sein muss. Für die anderen Parameter wird ein Standardwert festgelegt. Will man zum Beispiel einen Befehl, der die 1. binomische Formel oft nutzen, so kann man dies einfach als Makro schreiben:

Listing 4: binomisch Plus

```
\newcommand{\bp}[3][2]{\((#2 + #3) ^ #1\)}
\bp{x}{y}
```

$(x + y)^2$

In der Zeile `\newcommand{\bp}[3][2]{\((#2 + #3) ^ #1\)}` gibt

`\bp` den Namen des neuen Befehls an.

`[3]` ist die Anzahl der Parameter.

`[2]` ist der Standardwert des ersten Parameters, in diesem Fall 2.

`[2]` ist der Standardwert des ersten Parameters, in diesem Fall 2.

`\((#2 + #3) ^ #1\)` ist der ausgeführte Befehl.

Wenn nun der optinale Parameter auch gesetzt wird, kann der Exponent auch geändert werden:

Listing 5: binomisch Plus hoch 4

```
\bp[4]{32}{48}
```

$(32 + 48)^4$

`[4]` ist der Wert, mit dem der Standardwert des optinalen Parameters überschrieben wird.

Will man mehr als einen optionalen Parameter definieren, lässt sich dies mit dem Paket `xparse`[1] realisieren.

[1] **so:xparse**.

3 Bestehende Befehle überschreiben

Definiert man einen Befehl, der den gleichen Namen wie ein bereits vorhandener LaTeX-Befehl hat, wird eine Fehlermeldung in der Kompilierung des Dokuments angezeigt und der definierte Befehl funktioniert nicht. Wenn man wirklich einen bestehenden Befehl überschreiben will, kann man dies mit `renewcommand` erreichen.
Es ist zum Beispiel sinnvoll Befehle neu zu definieren, wenn man diese immer mit den gleichen Einstellungen braucht, und den Namen beibehalten will. In Präsentationen nutze ich `renewcommand` um den Befehl \footnotesize zu erneuern:

Listing 6: kleine Fußnoten

```
1 % Setze die größe der Fußnoten auf klein
2 \renewcommand{\footnotesize}{\small}
```

Der Syntax ist dem von \newcommand sehr ähnlich:

\footnotesize ist der Name des Befehls, der ersetzt oder erneuert werden soll.

\small ist der Befehl, durch den \footnotesize ersetzt wird.

In diesem Fall muss kein Wert über einen Parameter übergeben werden, \small ändert die Größe aller darauffolgenden Zeichen:

Listing 7: groß und klein

```
1 \large Groß \\
2 \small Klein
```

Groß
Klein

Es lassen sich Befehle ebenfalls so überschreriben, dass sie Parameter akzeptieren, diese können auch optional sein. Der Syntax ist der selbe, wie bei Makros mit optinalen Parametern.

4 Umgebungen definieren

Umgebungen sind Makros, die auf einen begrenzten Textbereich wirken. Sie werden mit den Befehlen \begin{...} und \end{...} eingerahmt. Neue (eigene) Umgebungen werden mittels \newenvironment definiert. Viele Änderungen innerhalb einer Umgebung werden am Ende der Umgebung aufgehoben.[2]

Listing 8: zentrierter Text

```
1 \begin{center}
2 Dieser Text ist zentriert
3 \end{center}
```

Dieser Text ist zentriert

\begin{center} ist der Anfang des `center`-Blocks.

„Dieser Text ist zentriert“ ist der zentrierte Text innerhalb des `center`-Blocks.

\end{center} ist das Ende des `center`-Blocks.

Zur Darstellung der Ergebnisse nach den Listings, verwende ich das folgende Makro (hier mit Zeillenumbrüchen):

Listing 9: Highlightbox

```
1 \newenvironment{hlbox}
2 {\begin{tcolorbox}[enhanced,colback=white,colframe=white,sharpish corners,
    fuzzy halo=0.5mm with lightgray]}
3 {\end{tcolorbox}}
```

Die Zeile \newenvironment{hlbox} gibt den Namen der neuen Umgebung an.

Die darauffolgende Zeile gibt den ersten Teil der Umgebung an, der mit \begin{hlbox} aufgerufen wird. Hier wird der erste Makroteil, der vor dem Text zur darstellung kommt definiert.

[enhanced,colb... sind die Optionen, die auf die `tcolorbox` angewendet werden.

{\end{tcolorbox}} bestimmt den Teil der Umgebung, der nach dem Text zur darstellung kommt.

[2] **wiki:environment**.

5 Bestehende Umgebungen überschreiben

Bestehende Umgebungen können mit dem Befehl \renewenvironment neu definiert werden.

```
\renewenvironment{bmatrix}
{ \begin{center} \begin{em} }
{ \end{em} \end{center} }

\begin{bmatrix}
Anstelle von einer Matrix ist dieser Text zentriert und hervorgehoben
\end{bmatrix}
```

Anstelle von einer Matrix in Klammern ist dieser Text zentriert und hervorgehoben

bmatrix ist der Name der Umgebung, die erneuert werden soll.

{ \begin{center} \begin{em} } ist der Teil, der vor dem eingesetzten Text kommt.

{ \end{em} \end{center} } ist der Teil des Makros, der nach dem eingesetzten Text kommt und die Umgebung beendet.

6 Nachtrag

Hier ein textumflossendes Bild einer Katze:

Abbildung 1: Katze

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Listings

Abbildungsverzeichnis