

# Das faktor-Package – Faktor-Ringe, -Gruppen, -Räume\*

Paul Ebermann<sup>†</sup>

11. April 2006

## Zusammenfassung

In der Algebra, der Topologie, etc. tauchen gelegentlich Faktorstrukturen auf – etwa Faktormengen, Faktorgruppen, Faktorringe, Faktorräume. Diese haben alle eine ähnliche Notation:  $A/B$ . Bisher konnte man dies in  $\text{\LaTeX}$  nicht vernünftig setzen, dieses Paket schafft Abhilfe.

## Inhaltsverzeichnis

<b>1 Benutzerdoku</b>	<b>1</b>
1.1 Anwendung . . . . .	1
1.1.1 Beispiel . . . . .	2
1.2 Abhängigkeiten . . . . .	2
<b>2 Implementation</b>	<b>2</b>
<b>3 Bug</b>	<b>3</b>
<b>4 Liste der Änderungen</b>	<b>4</b>
<b>5 Index</b>	<b>4</b>

## 1 Benutzerdoku

### 1.1 Anwendung

Um das Paket zu laden, verwendet man wie üblich

```
\usepackage{faktor}
```

Package-Optionen gibt es bisher nicht.

`\faktor` Das namensgebende (und zur Zeit einzige) Makro ist `\faktor`. Es funktioniert nur im mathematischen Modus.

---

\*Dieses Dokument gehört zu faktor v0.1b, vom 2006/04/05.

<sup>†</sup>Paul.Ebermann@gmx.de

`\faktor[⟨stylecmd⟩]{⟨zähler⟩}{⟨nenner⟩}`

⟨zähler⟩ und ⟨nenner⟩ sollten jeweils Formeln für den mathematischen Modus sein. Dieses Makro setzt dann ein Faktor-Objekt, also „⟨zähler⟩ faktorisiert nach ⟨nenner⟩“. Im Effekt wird dabei ein  $\diagup$  auf natürlicher Höhe gemalt, links daneben (höhergesetzt) ⟨nenner⟩, rechts daneben (tiefergesetzt) ⟨zähler⟩.

Falls ⟨stylecmd⟩ angegeben wurde, wird es auf ⟨zähler⟩ und ⟨nenner⟩ jeweils bei der Ausgabe angewandt – dies könnte also etwa ein Kommando sein, um Schriftart o.ä. zu ändern.

### 1.1.1 Beispiel

`$$\faktor{\sum_{i=1}^n k[X]}{\sum_{i=1}^n k[X] \cdot \theta_i}$$`

ergibt:

$$\sum_{i=1}^n k[X] \diagup \sum_{i=1}^n k[X] \cdot \theta_i$$

## 1.2 Abhängigkeiten

`\diagup` Dieses Package verwendet das Zeichen `\diagup` ( $\diagup$ ) aus dem `amssymb`-Paket – wenn dieses nicht definiert ist, gibt es also einen Fehler, sobald `\faktor` verwendet wird. Daher sollte dieses Paket nur gemeinsam mit `amssymb` oder einem anderen Paket, welches ein gleich benanntes und ähnlich funktionierendes Zeichen zur Verfügung stellt, verwendet werden. Notfalls kann man dieses Zeichen auch einzeln laden (eine Anleitung dafür findet man in der Dokumentation von `amsfonts`), oder als `picture` simulieren.

## 2 Implementation

Die Implementation ist ähnlich zu der des `nicefrac`-Paketes, allerdings etwas vereinfacht, weil wir ja nichts verkleinern wollen.

1 `*package`

`\faktor@zaehlerhoehe` In diesen beiden Registern speichern wir nachher die Höhen von „Zähler“ und  
`\faktor@nennerhoehe` „Nenner“ (d.h. Grundobjekt und faktorisiertes Objekt).

2 `\newlength{\faktor@zaehlerhoehe}`

3 `\newlength{\faktor@nennerhoehe}`

`\faktor` Der erste Parameter ist optional, vorgabemäßig leer. `\faktor` ruft `\faktor@impl@` auf, und gibt mittels `\mathpalette`, je nach aktuellem Mathe-Stil, das Kommando, um in diesen Stil zu wechseln (also `\displaystyle`, `\textstyle`, ...), mit. Die `{}` drumherum sorgen dafür, dass das ganze als ein einziges Objekt gewertet wird, an das man z.B. Sub- und Superscript anhängen kann. (Außerdem

haben wir eine Gruppe, d.h. die Zuweisungen zu `\faktor@zaehlerhoehe` und `\faktor@nennerhoehe` bleiben lokal.

```

4 \DeclareRobustCommand*\faktor}[3] []
5 {
6   { \mathpalette{\faktor@impl@}{#1}{#2}{#3}} }
7 }

```

`\faktor@impl@` Dieses Hilfsmakro entsorgt einfach nur die zusätzlichen Klammern, welche von `\mathpalette` um `#2` eingefügt wurden, damit `\faktor@impl` vier Argumente und nicht nur zwei bekommt.

```
8 \newcommand*\faktor@impl@[2]{\faktor@impl#1#2}
```

`\faktor@impl` `{\modeCmd}{\zusatz}{\zähler}{\nenner}`

`\faktor@impl` macht die eigentliche Arbeit. Es ermittelt zunächst Höhen von `\zähler` und `\nenner` (abhängig von `\modeCmd` und `\zusatz`).

```

9 \newcommand*\faktor@impl}[4]{
10   \settoheight{\faktor@zaehlerhoehe}{\ensuremath{#1#2{#3}}}%
11   \settoheight{\faktor@nennerhoehe}{\ensuremath{#1#2{#4}}}%

```

Dann wird, um die Hälfte seiner Höhe nach oben geschoben, `\zähler` gesetzt.

```
12   \raisebox{0.5\faktor@zaehlerhoehe}{\ensuremath{#1#2{#3}}}%

```

Jetzt kommt ein negativer horizontaler Abstand, der diagonale Strich, dann noch ein negativer horizontaler Abstand, um das ganze etwas zusammenzurücken.

```
13   \mkern-5mu\diagup\mkern-4mu%
```

Schließlich kommt `\nenner`, um die Hälfte seiner Höhe nach unten geschoben.

```

14   \raisebox{-0.5\faktor@nennerhoehe}{\ensuremath{#1#2{#4}}}%
15 }

```

Das war es.

```

16 \endinput
17 \endpackage

```

### 3 Bug

Beim höher- und tiefersetzen von Zähler und Nenner wird jeweils nur die *Höhe* des Objektes berücksichtigt, nicht die Tiefe (d.h. Unterlängen). Dadurch (?) kommt es bei größeren Brüchen gelegentlich zu etwas komischem Aussehen:

$$a/b/b/c$$

## 4 Liste der Änderungen

v0.1		Quelltext präzisiert. . . . .	1
	Allgemein: Erste Fassung als .DTX		1
	<code>\faktor</code> : Jetzt Berücksichtigung	v0.1b	
	des Mathe-Stils. . . . .	Allgemein: Tippfehler in der Doku-	
		mentation beseitigt. . . . .	1
v0.1a			
	Allgemein: Lizenzbestimmungen im		

## 5 Index

Schräggedruckte Nummern verweisen auf die Seite, auf der der Eintrag beschrieben ist, unterstrichene Nummern zeigen auf die Zeilennummer der Definition, sonstige Zahlen auf die Zeilennummer einer Verwendung.

<b>D</b>		<b>M</b>	
<code>\diagup</code> . . . . .	2, 13	<code>\mathpalette</code> . . . . .	6
		<code>\mkern</code> . . . . .	13
<b>F</b>		<b>R</b>	
<code>\faktor</code> . . . . .	1, <u>4</u>	<code>\raisebox</code> . . . . .	12, 14
<code>\faktor@impl</code> . . . . .	8, <u>9</u>		
<code>\faktor@impl@</code> . . . . .	6, <u>8</u>	<b>S</b>	
<code>\faktor@nennerhoehe</code> . . . . .	<u>2</u> , 11, 14	<code>\settoheight</code> . . . . .	10, 11
<code>\faktor@zaehlerhoehe</code> . . . . .	<u>2</u> , 10, 12		