

Das extpfeil-Package – Weitere erweiterbare Pfeile*

Paul Ebermann[†]

23. März 2006

Inhaltsverzeichnis

1 Benutzerdoku	1
1.1 Generelle Benutzung der Pfeile	1
1.2 Die einzelnen Pfeile	2
2 Implementation	2
2.1 geladene Pakete	2
2.2 Die verwendeten <code>amsmath</code> -Funktionen	2
2.3 Neue Pfeile	3
2.4 Weitere Pfeile	4
3 Schluss	4
4 Liste der Änderungen	4
5 Index	5

1 Benutzerdoku

Dieses Paket definiert eine Reihe erweiterbare Pfeile, sowie ein Kommando, um selbst weitere zu definieren (siehe Abschnitt 2.3).

1.1 Generelle Benutzung der Pfeile

Die Pfeile funktionieren wie die Pfeile `\xleftarrow` und `\xrightarrow` aus `AMS-Math` sowie diverse weitere Pfeile aus `mathtools` bzw. `extarrows` (letzteres Paket habe ich erst später entdeckt – es hat auch keine Dokumentation). Die generelle Syntax ist

$$\langle pfeilname \rangle [\langle subscript \rangle] \{ \langle superscript \rangle \}$$

*Diese Dokumentation gehört zu `extpfeil` v0.2, vom 2006/03/23.

[†]`Paul-Ebermann@gmx.de`

$\langle pfeilname \rangle$ ist das entsprechend definierte Kommando (siehe Abschnitt 1.2) $\langle subscript \rangle$ eine Formel, die (falls vorhanden) unter den Pfeil gesetzt wird, $\langle superscript \rangle$ eine, die über den Pfeil gesetzt wird. Der Pfeil selbst nimmt dann eine ausreichend hohe Länge an, um an beiden Seiten etwas über die Formel hinauszuragen.

1.2 Die einzelnen Pfeile

<code>\xtwoheadrightarrow</code>	<code>\xtwoheadrightarrow[a]{b}</code>	$\xrightarrow[a]{b}$	ein einfacher Pfeil nach rechts mit Doppelspitze
<code>\xtwoheadleftarrow</code>	<code>\xtwoheadleftarrow[a]{b}</code>	$\xleftarrow[a]{b}$	ein einfacher Pfeil nach links mit Doppelspitze
<code>\xlongequal</code>	<code>\xlongequal[a]{b}</code>	$\xlongequal[a]{b}$	kein Pfeil, sondern einfach nur ein verlängerbares Gleichheitszeichen.

2 Implementation

1 $\langle *package \rangle$

2.1 geladene Pakete

Zunächst laden wir das `amsmath`-Package, in welchem die verwendeten Makros definiert sind, sowie `amssymb`, in dem sich einige verwendete Symbole aufhalten.

```
2 \RequirePackage{amsmath}
3 \RequirePackage{amssymb}
```

2.2 Die verwendeten `amsmath`-Funktionen

Leider ist die Dokumentation der \mathcal{AMS} -Math-Funktionen¹ etwas dürftig – ich musste mir selbst zusammenreimen, was wie funktionieren könnte. Deswegen sind im folgenden die Ergebnisse meiner Forschungen niedergelegt.

<code>\relbar</code>	Ich verwende für meine Pfeile <code>\relbar</code> – das ist einfach nur ein Minus-Zeichen
<code>\Relbar</code>	($-$) (der Leerraum oben und unten etwas verkleinert, damit Super- und Subskripte nicht so weit weg sind), als verlängerbares Element. Genauso gibt es auch <code>\Relbar</code> ($=$) für Doppelpfeile.
<code>\arrowfill@</code>	<code>\arrowfill@{<links>}{<mitte>}{<rechts>}{<größe>}</code> wird benutzt, um dehnbare Pfeile zu erstellen. Das Makro bekommt 4 Parameter: $\langle links \rangle$ ist das linke Ende des Pfeils, $\langle rechts \rangle$ das rechte Ende. $\langle mitte \rangle$ ist das, was verlängerbar gemacht wird - also am besten <code>\relbar</code> oder <code>\Relbar</code> . $\langle größe \rangle$ wird dem Makro von <code>\ext@arrow</code> (bzw. anderen Makros, die mit <code>\arrowfill@</code> gebastelte Pfeile verwenden) übergeben, darum muss man sich beim

¹in Abschnitt 12 von `amsmath.dvi`, ab Seite 25 in der Version 2.13

Erstellen von Pfeilen also nicht kümmern – es ist ein Kommando, um die Mathe-Schriftgröße einzustellen, (bei `\ext@arrow` ist es `\displaystyle`).

Ein Beispiel der Anwendung² ist

```
\def\leftarrowfill@{\arrowfill@\leftarrow\relbar\relbar}
```

Das linke Ende ist also ein `\leftarrow` (\leftarrow), das rechte Ende (genauso wie die Mitte) ein `\relbar` ($\bar{-}$). Insgesamt führt das (mit einer Übergabe von `\textstyle`) zu \leftarrow (hier ziemlich lang, da es ja den Platz füllt).

`\ext@arrow` `\ext@arrow{<li 1>}{<re 1>}{<li 2>}{<re 2>}{<Pfeil>}{<oben>}{<unten>}` ist nun das Kommando, welches einen (mit `\arrowfill@` erstellten) Pfeil in der richtigen Länge mit etwas darüber und darunter ausgibt.

`<oben>` ist eine Formel, die über dem Pfeil ausgegeben wird, `<unten>` eine Formel, die unter dem Pfeil ausgegeben wird. Beides kann auch leergelassen werden, dann wird eben nichts dort ausgegeben.

`<Pfeil>` gibt den Pfeil an, der zu verwenden ist – dies sollte aus `\arrowfill@` mit den ersten drei Parametern gebildet sein (entweder in `{}` – so werden wir es nachher machen – oder als extra-Makro (wie `\leftarrowfill@`).

`<li 1>` und `<re 1>` geben an, wieviel Platz vor und nach `<oben>` und `<unten>` gelassen werden soll – mittels `\mkern`, die Einheit dazu ist μ (*Math units*). Damit kann man die Position der Formeln über/unter dem Pfeil etwas seitlich anpassen.

`<li 2>` und `<re 2>` funktionieren ähnlich, werden allerdings nur zum Berechnen der Länge des Pfeiles verwendet. Damit kann man die Länge des Pfeiles noch etwas über die Breite der Formel hinaus ausdehnen. Ich habe keine Ahnung, ob zwischen `<li 2>` und `<re 2>` ein funktioneller Unterschied besteht, oder ob einfach nur die Summe von beidem zählt.

Zum Definieren eines Pfeiles mit Sub- und Superskript geht man z.B. so vor³:

```
\newcommand{\xleftarrow}[2] []{%
\ext@arrow 3095\leftarrowfill@{#1}{#2}}
```

Die ersten 4 Parameter werden also hier als einzelne Ziffern übergeben (ohne `{}`), dann (`\leftarrowfill@`) ein mit `\arrowfill@` erstellter dehnbare Pfeil, zum Schluss die beiden Parameter des definierten Makros. Das Ergebnis ist z.B.

$$\overset{\textit{oben}}{\leftarrow} \underset{\textit{unten}}{\quad} .$$

2.3 Neue Pfeile

Wir kombinieren jetzt die beiden oben vorgestellten Kommandos, um so einfacher eigene Pfeile erstellen zu können.

`\newextarrow` `\newextarrow{<befehl>}{<platz>}{<pfeil>}` definiert das Kommando `<befehl>` als dehnbare Zeichen (mit ober- und Unterscript), mit den Platz-Angaben in `<platz>` (dies sollte 4 Token enthalten) und den Pfeil-Teilen in `<pfeil>` (das sollte drei Token enthalten).

²aus `amsmath.dtx`

³aus `amsmath.dtx`, Zeilenumbruch von mir

`\newextarrow` Das damit definierte Makro ruft also `\ext@arrow` auf, mit den Parametern $\langle platz \rangle$, dem aus `\arrowfill@` und $\langle pfeil \rangle$ zusammengesetzten Pfeil sowie seinen beiden eigenen Parametern (der erste davon ist optional) auf.

```
4 \newcommand*\newextarrow[3] {%
5   \newcommand*{#1}[2] [] {\ext@arrow #2{\arrowfill@#3}{##1}{##2}}
6 }
```

2.4 Weitere Pfeile

Ich verwende jetzt `\newextarrow`, um weitere Pfeile zu erstellen.

`\xtwoheadrightarrow` Unser Pfeil sollte ziemlich weit überstehen, deswegen die `{40}`. 05 scheint zur Justierung der Formeln ganz gut zu passen.

```
7 \newextarrow{\xtwoheadrightarrow}
8   {05{40}0}
```

Den Pfeil selbst setzen wir aus \rightarrow (aus `amssymb`) sowie den üblichen `\relbar` zusammen:

```
9   {\relbar\relbar\xtwoheadrightarrow}
```

`\xtwoheadleftarrow` Das geht analog zu `\xtwoheadrightarrow` (die $\langle li \rangle$ - und $\langle re \rangle$ -Parameter sind vertauscht, da der Pfeil ja genau gespiegelt ist).

```
10 \newextarrow{\xtwoheadleftarrow}
11   {500{40}}
12   {\twoheadleftarrow\relbar\relbar}
```

`\xmapsto` Hier noch eine verlängerbare Variante von \mapsto und \mapsto . `mathtools` hat bereits eine Implementation davon, aber diese ist etwas länger und gefällt mir daher besser.

```
13 \let\xmapsto\undefined
14 \newextarrow{\xmapsto}{0599}{\mapstochar\relbar}\relbar\rightarrow}
```

`\xlongequal` Alle drei Teile bestehen aus $=$,

```
15 \newextarrow{\xlongequal}{5599}{\Relbar\Relbar\Relbar}
```

3 Schluss

Das war es fürs erste – später kommen bestimmt noch mehr Pfeile hinzu.

```
16 \endinput
17 \</package>
```

4 Liste der Änderungen

v0.0	v0.1
Allgemein: Anfang der Fassung als DTX	Allgemein: Alle Pfeile verwenden nun <code>newextarrow</code> zur Definition.
1	3

<code>\newextarrow</code> : Neu hinzu	3	<code>\xlongequal</code> : Neu hinzu	4
		v0.2	
		<code>\xmapsto</code> : Neu.	4

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.

	A		<code>\relbar</code>	2, 9, 12, 14
<code>\arrowfill@</code>	2, 5		<code>\rightarrow</code>	14
	E		T	
<code>\ext@arrow</code>	2, 5		<code>\twoheadleftarrow</code>	12
	L		<code>\twoheadrightarrow</code>	9
<code>\let</code>	13		U	
	M		<code>\undefined</code>	13
<code>\mapstochar</code>	14		X	
	N		<code>\xlongequal</code>	1, <u>15</u>
<code>\newextarrow</code>	3, <u>4</u> , 7, 10, 14, 15		<code>\xmapsto</code>	<u>13</u>
	R		<code>\xtwoheadleftarrow</code>	1, <u>10</u>
<code>\Relbar</code>	2, 15		<code>\xtwoheadrightarrow</code>	1, <u>7</u>