The $\varphi o rmula$ style

Version 0.2

Andreas Tille

April 29, 1998

Contents

1 Usage **1**

2 Definitions of $\varphi o rmula$ **2**
2.1 Definining single $\varphi o rmula$ commands 2
2.2 Formelzeichen mit Anhang 3
2.3 Formelzeichen mit Einschub 4
2.4 Differenzenquotienten 5
2.5 Maßeinheiten 5
2.6 How to get all $\varphi o rmula$ definitions 6
2.7 Einfache Differenzenquotienten 6

3 Language support **7**

Abstract

$\varphi o rmula$ is inteded to be helpful when using symbols inside and outside math-mode. The symbols which can be defined using $\varphi o rmula$ should help to ensure a unique layout over all documents using the same definitions.

Additionally there is support for physical units. The use of $\varphi o rmula$ units makes sure to have the right spacing.

1 Usage

To use $\varphi o rmula$ you have to include

```
\usepackage{formula}
```

in the preamble of your LaTeX2e-document. The `formula.sty` depends from some other packages which are:

- `xspace.sty` from *David Carlisle*,
- `amsfonts.sty` and `amstext.sty` contained in the $\mathcal{AMS}$-LaTeX-package and
- `textcomp.sty` from *Jörg Knappen* (at least Version 1998/03/05 v1.9n; if you really havn't access to this Version you will have to replace `\textcelsius` by `\textcentigrade` in `formula.dtx`).

Loading of `formula.sty` takes some time. If the predefined commands are not used this time can be shortened by using the option `[nopredefinition]`. To get a list of all predefined commands in $\mathcal{F}ormula$ use prepverz (see section 2.6).

2 Definitions of $\mathcal{F}ormula$

2.1 Definining single $\mathcal{F}ormula$ commands

Zur Definition von einfachen Symbolen wird `\formuladef` verwendet. Dieses Makro ist mit vier Argumenten aufzurufen.

`\formuladef {`*Name*`} {`*Inhalt*`} {`*Beschreibung*`} {`*Artikel*`}`

Dabei steht *Name* für den Namen eines zu definierenden Makros das im Folgenden mit `\Name` verwendet werden kann. Das Makro `\Name` bewirkt, daß *Inhalt* im Mathemodus gesetzt wird, unabhängig davon, ob der Mathemodus oder der Textmodus aktiv ist. Falls der *Inhalt* oder Teile desselben in Roman gesetzt werden sollen, so ist das durch `\mathrm{`*Inhalt in Roman*`}` zu kennzeichnen. Durch `\formuladef` wird ein weiteres Makro `\Namedoc` erzeugt. Dieses Makro hat folgenden Inhalt:

`\Name &` *Beschreibung* `\\`

Der Zweck dieses Makros ist die Vereinfachte Schreibweise für ein Abkürzungsverzeichnis. Ein typisches Abkürzungsverzeichnis sieht folglich so aus:

```
\begin{tabular}{ll}
\einsdoc
\zweidoc
\dreidoc
\end{tabular}
```

wobei `\eins`, `\zwei` und `\drei` durch `\formuladef` definierte Befehle sein mögen. Zwei weitere durch `\formuladef` erzeugte Makros sind `\Nametxt` und `\Nameart`. Sie stellen das dritte bzw. vierte Argument des Aufrufs von `\formuladef` bereit. Diese beiden Befehle können sich bei der Erläuterung von Symbolen in einer Formel als nützlich erweisen. Der *Artikel* kann natürlich auch `{}` sein. Er wird lediglich deshalb extra aufgelistet, weil er im

Abkürzungsverzeichnis in der Regel nicht benötigt wird, bei Erläuterungen von Formelzeichen jedoch erst einen vollständigen Satz ermöglicht. Das letzte durch `\formuladef` erzeugte Makro heißt `\Namemy`. Es gibt dem Autor die Möglichkeit, einen Überblick über seine 1001 `\formuladef`-Definitionen zu behalten. Es hat folgenden Inhalt:

```
\texttt{\Name} & \Namedoc
```

Damit wird für den Autor dokumentiert, welchen Befehl er einzugeben hat und was sich hinter dem Befehl verbirgt. Beispiel:

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[nopredefinition]{formula}

\pagestyle{empty}
\textheight 55\baselineskip
\advance\oddsidemargin by -10mm
\textwidth 150mm

\formuladef IQ  {\text{IQ}}  {Intelligenzquotient}  {der}
\formuladef Kl  {K}          {Klugheit}             {die}
\formuladef Du  {D}          {Dummheit}             {die}
\begin{document}
\section*{Wie dumm ist der Zufall}

Herr Rainer Zufall hat einen \IQ von \texttt{???} {(\em Zensiert,
Frau Emma N. Z. Daten-Schutz)}. Dabei ist der \IQ definiert als
\beq
   \IQ = \frac{\Kl}{\Du} \label{IQ}
\eeq
wobei \Kl \Klart \Kltxt des Herrn Zufall und \Du \Duart \Dutxt
dieses werten Herrn ist. \par
```

Bemerkung: `\formuladef` und auch die im Folgenden definierten Befehle werden *global* definiert, daß heißt, auch wenn die Definitionen der Befehle innerhalb einer Umgebung vereinbart werden, sind sie im gesamten folgenden Text bekannt.

2.2 Formelzeichen mit Anhang

Es kann vorkommen, daß an ein Formelzeichen etwas angehängt werden soll (Indizes, Potenzen, etc.). Dafür wurde eine Abwandlung von `\formuladef` eingeführt:

`\formulaarg` {*Name*} {*Inhalt*} {*Beschreibung*} {*Artikel*} {*Argument*}

Die ersten vier Argumente von `\formulaarg` entsprechen denen von `\formuladef`. Hinzugekommen ist lediglich das fünfte Argument. Es ist nur in `\Namedoc` von Bedeutung: In der Dokumentation wird das *Argument* an den Befehl angehängt. In der *Beschreibung* ist zweckmäßigerweise auf diese Argument Bezug zu nehmen. Das sieht zum Beispiel so aus:

```
\formulaarg {Kli} {K_} {Klugheit bei der T\"atigkeit $X$} {die} {X}
\formulaarg {Dui} {D_} {Dummheit bei der T\"atigkeit $X$} {die} {X}

Angenommen, Herr Zufall stellt sich bei verschiedenen T\"atigkeiten
unterschiedlich dumm an, so l\"a\ss{}t sich (\ref{IQ}) folgenderma\ss{}en
pr\"azisieren:
\beq \label{IQi}
   \IQ = \frac{\sum\limits_{i=1}^M\Kli{i}}{\sum\limits_{i=1}^N\Dui{i}}
\eeq\par
```

2.3 Formelzeichen mit Einschub

Auch wenn ein Argument zwischen Teile eines zu definierenden Befehls eingebaut werden soll, wird eine Definition bereitgestellt:

`\formulamit` {*Name*} {*I1*} {*I2*} {*Beschreibung*} {*Artikel*} {*Argument*}.

In `\formulamit` wird offensichtlich der *Inhalt* in zwei Teilen *I1* und *I2* angegeben. Zwischen diese beiden Teile wird das Argument eingefügt. Das läßt sich auch wieder am Beispiel eines reinen Zufalls demonstrieren:

```
\formulamit {KlT} {K_} {^T} {Klugheit am Tag} {die} {X}
\formulamit {KlN} {K_} {^N} {Klugheit in der Nacht} {die} {X}
\formulamit {DuT} {D_} {^T} {Dummheit am Tag} {die} {X}
\formulamit {DuN} {D_} {^N} {Dummheit in der Nacht} {die} {X}

Wird nun noch in Betracht gezogen, da\ss{} der Zufall im Traum eine ganz
andere Rolle spielt, als im realen Leben, so kann zwischen den Klugheiten
bei Tag (\KlT{X}) und Nacht (\KlN{X}), sowie den Dummheiten zu diesen
Tageszeiten (\DuT{X} bzw. \DuN{X}) unterschieden werden. Bei Reiner
Zufall ist die Klugheit tags\"uber vernachl\"assigbar klein gegen\"uber
der beim Tr\"aumen. Die Dummheit verh\"alt sich genau umgekehrt.

Also ergibt sich aus (\ref{IQi})
\beq
   \IQ \simeq \frac{\sum\limits_{i=1}^M\KlN{i}}{\sum\limits_{i=1}^N\DuT{i}}.
\eeq
```

2.4 Differenzenquotienten

Schließlich wird noch ein Makro bereitgestellt, mit der Differenzenquotienten leicht definiert werden können. Dieses wird folgendermaßen aufgerufen:

`\formuladiff` {*Name*} {*Art*} {*y*} {*x*} {*Beschreibung*} {*Artikel*}.

Dabei haben *Name*, *Beschreibung* und *Artikel* die gleiche Bedeutung wie in den vorangegangenen Definitionen. Das Argument *Art* kann den Wert 'd' oder '∂' haben, kennzeichnet also, ob es sich um eine totale oder partielle Ableitung handeln soll. Die abhängige Variable y erscheint im Nenner, die unabhängige Variable x im Zähler. Einem auf diese Art und Weise definierten Befehl muß ein Argument folgen, daß den Grad der Ableitung angibt. Mit diesem Argument wird im Nenner das entsprechende Differentialzeichen (*Art*) potenziert, im Zähler dagegen die unabhängige Variable (x) – wie das halt so üblich ist.

```
\formuladiff Klt {d} {\Kl} {t} {Klugheitszuwachs n-ten Grades} {der}

Da Herr Zufall ein eifriger Leser der Boulevardpresse ist, gilt:
\bea
  \Klt{}  & < & 0 \qquad\forall\quad t > t_0 \\
  \Klt{2} & > & 0 \qquad\forall\quad t > t_0
\eea
```

2.5 Maßeinheiten

Zur Vereinheitlichung von Maßeinheiten wurde der Befehl `\formulaunit` eingeführt. Er hat folgende Syntax:

`\formulaunit` {*Name*} {*rm*} {*gr*} {*Einheit*} {*Einheit in Worten*}.

Wie in den Definitionen oben ist *Name* ein zu definierender Befehl. Weiterhin werden die Befehle `\Namedoc`, `\Nametxt` und `\Namemy` erzeugt, die die gleiche Bedeutung wie oben haben. Die Argumente *rm* und *gr* sind als Vorsätze vor Einheiten zu gebrauchen. Dabei steht *rm* für einen Vorsatz aus lateinischen Buchstaben (z.B. m für Milli) und *gr* für einen Vorsatz in griechischen Buchstaben (z.B. µ für Mikro). Die Definitionen

```
\formulaunit mm {m} {} m {Millimeter}
\formulaunit mum {} {\mu} m {Mikrometer}
```

bewirken folglich, daß die Eingabe

```
1\mm = 1000\mum
```

im Ergebnis $1\,\mathrm{mm} = 1000\,\mu\mathrm{m}$ liefert. Zu beachten ist, daß der Einheit-Befehl immer unmittelbar auf den Zahlenwert folgt, damit nicht zusätzlicher Leerraum eingefügt wird und TEX an dieser Stelle nicht trennt. *Einheit* ist die Einheit ohne weitere Vorsätze. Einheiten werden unabhängig davon, ob sie im Mathemodus oder nicht verwendet werden, immer in Roman gesetzt. Der Wert der Einheit wird durch `\,` von der Einheit getrennt.

2.6 How to get all $\varphi_o rmula$ definitions

$\varphi_o rmula$ contains a number predefined commands. The script `prepverz` can be used to get an overview of this predefinitions and to remember your own $\varphi_o rmula$ definitions. There should be no problem on any `UNIX` system. Using `DOS` or `OS/2` (rename `prepverz.bat` to `prepverz.cmd` in the latter case) you need a working `grep` and `sed`. There should be no problems with any `grep` version but there are reports of `DOS` `sed`s which failed. The `GNU-sed` version 2.05 is known to work reliable. Try to get a copy of it for your operating system.

The streameditor `sed` is used to get all `\formula...`-definitions from any input file of `prepverz`. These definitions where written into a `longtable` environment in the outputfile `abverz.tex`. For example

```
prepverz formula.sty
```

creates `abverz.tex` with all predefined $\varphi_o rmula$ definitions. Use your own files containing $\varphi_o rmula$ definitions to get all your own defined commands. Then you can type

```
latex abverz.tex
```

twice (to get the longtable formatted correctly) and you will get a fairly formated output of the definitions.

2.7 Einfache Differenzenquotienten

Es kann auch einmal der Fall eintreten, daß der Nutzer für eine Ableitung keinen extra Befehl definieren will. Es gibt deshalb für partielle Ableitungen (wenn auch totale gewünscht werden, kann das schnell analog definiert werden) eine kleine Unterstützung.

Eingabe	Resultat	Beschreibung
`\odif{`y`}{`x`}`	$\frac{\partial y}{\partial x}$	1. Ordnung ohne Klammern
`\pdif{`y`}{`x`}`	$\left(\frac{\partial y}{\partial x}\right)$	1. Ordnung mit Klammern
`\osdif{`y`}{`x`}`	$\frac{\partial^2 y}{\partial x^2}$	2. Ordnung nach einer Variablen ohne Klammern
`\oodif{`z`}{`y`}{`x`}`	$\frac{\partial^2 z}{\partial y \partial x}$	2. Ordnung nach zwei Variablen ohne Klammern
`\ppdif{`z`}{`y`}{`x`}`	$\left(\frac{\partial^2 z}{\partial y \partial x}\right)$	2. Ordnung nach zwei Variablen mit Klammern
`\ooodif{`u`}{`z`}{`y`}{`x`}`	$\frac{\partial^3 u}{\partial z \partial y \partial x}$	3. Ordnung ohne Klammern
`\pppdif{`u`}{`z`}{`y`}{`x`}`	$\left(\frac{\partial^3 u}{\partial z \partial y \partial x}\right)$	3. Ordnung mit Klammern
`\oosdif{`z`}{`y`}{`x`}`	$\frac{\partial^3 z}{\partial y^2 \partial x}$	3. Ordnung ohne Klammern, wobei nach einer Variablen zweimal abgelitten wird

Offensichtlich ist diese Tabelle unvollständig und recht willkürlich zusammengestellt. Es wurden nur diejenigen Fälle aufgenommen, die in der bisherigen Praxis auftraten. Analoge Definitionen können leicht abgeleitet werden.

`\buildul` Mit `\buildrel` können im Mathematiksatz neue mathematische Relationen durch Übereinandersetzen von verschiedenen Symbolen erzeugt werden.

`\buildrel {`*oben*`} \over {`*unten*`}`

Dabei wird `{`*oben*`}` verkleinert über `{`*unten*`}` gesetzt. Ein Pendant zu `\buildrel` wird durch $\varphi_o rmula$ bereitgestellt. Mit dem Befehl `\buildrul` wird das zweite Argument verkleinert unter das erste gesetzt:

`$\buildrul\longrightarrow\over {`i`\to\infty}$` $\buildrul\longrightarrow\over {i\to\infty}$

3 Language support

Originally $\varphi_o rmula$ was designed for internal use and the documentation is mostly in German. This will change in the future but may take a little time. For users who didn't understand German all comments in the code section are written in English and hopefully they are understandable and useful.

To give easy support for translations the macro `\eorg` was included. It is intended to define `\formuladef` and friends macros with German and English description (the `\*doc` part of the created macro. For example you can use

`\formuladef` Du `{`D`}` `{\eorg{`*foolery*`}{`*Dummheit*`}}` `{\eorg{`*the*`}{`*die*`}}`

Questions, comments or additions to:

Andreas Tille
Gartenstraße 8
D-38855 Wernigerode

Martin-Luther-Universität Halle/Wittenberg
Fachbereich Physik
Experimentalphysik II
Friedemann-Bach-Platz 6
D-06108 Halle / Saale
Telefon: (+49 345) 55 25550
Fax: (+49 345) 55 27 158

E-Mail: `tille@physik.uni-halle.de`
WWW: `http://www.physik.uni-halle.de/ e2od5`