

ANWENDER- DOKUMENTATION	Teil 3 "Setzen mathematischer Ausdrücke mit neqn 11/87	AC A 7100/7150
MOS		MUTOS 1700

Programmtechnische
Beschreibung

Teil 3
"Setzen mathematischer
Ausdrücke mit neqn
11/87

VEB Robotron-Projekt Dresden

Ausgabe: MOS

Die Ausarbeitung dieser Dokumentation erfolgte durch ein Kollektiv des VEB Robotron-Elektronik Dresden Stammbetrieb des VEB Kombinat Robotron.

Nachdruck und jegliche Vervielfältigung, auch auszugsweise, sind nur mit Genehmigung des Herausgebers zulässig.

Im Interesse einer ständigen Weiterentwicklung werden alle Leser gebeten, Hinweise zur Verbesserung der Dokumentation dem Herausgeber mitzuteilen.

Herausgeber:

VEB Robotron-Projekt Dresden
Leningrader Str. 9
Dresden 8010

(C) VEB Kombinat Robotron

Kurzreferat

Die vorliegende Schrift beschreibt die Anwendung des Programms neqn für die Formatierung mathematischer Ausdrücke in Dokumenten und erläutert die Benutzung von Schlüsselworten zur Darstellung mathematischer Ausdrücke an Beispielen.

Inhaltsverzeichnis	Seite
1. EINLEITUNG	
2. ELEMENTE VON NEQN	
2.1. HERVORHEBEN VON GLEICHUNGEN	2-1
2.2. ABKÜRZUNGEN FÜR IN-LINE GLEICHUNGEN	2-1
2.3. BEHANDLUNG VON LEERZEICHEN BEI DER EIN- UND AUSGABE	2-2
2.4. INDIZES	2-3
2.5. BRÜCHE	2-4
2.6. QUADRATWURZELN	2-4
2.7. SPEZIELLE ZEICHEN	2-5
2.8. ÄNDERUNGEN DER SCHRIFTART	2-5
2.9. UNTERSCHIEDSZEICHEN	2-6
2.10. APOSTROPHIEREN VON TEXT	2-7
2.11. AUSRICHTEN VON GLEICHUNGEN	2-7
2.12. KLAMMERN	2-8
2.13. SPALTEN	2-8
2.14. MATRIZEN	2-9
2.15. DEFINITIONEN	2-9
2.16. HORIZONTALE UND VERTIKALE BEWEGUNGEN	2-10
2.17. VORRÄNGE DER SCHLÜSSELWORTE	2-11
3. FEHLERMELDUNGEN	
4. BENUTZUNG IM SYSTEM MUTOS	

1. Einleitung

~~~~~

Diese Schrift beschreibt die Anwendung des Programms `$fat neqn$` für die Formatierung mathematischer Ausdrücke in Dokumenten. Die Textformatierung mathematischer Ausdrücke durch traditionelle Methoden verlangt umständliche manuelle Operationen, denn mathematische Ausdrücke haben ihre eigenen typographischen Konventionen, die sich von gewöhnlichem Text stark unterscheiden. Eine zweite Schwierigkeit ist der zweidimensionale Charakter von mathematischen Ausdrücken durch hoch- und tiefgestellte Indizes, Zeilenziehung, zusammengesetzten Zeichen, wie Klammern und Wurzelzeichen und verschiedenen Positionierungsproblemen.

Die Sprache für das Programm `$fat neqn$` ist so konzipiert, daß sie von den Nutzern keine speziellen Kenntnisse über mathematische Ausdrücke und Textformatierung verlangt.

Das Programm `$fat neqn$` arbeitet als ein Präprozessor für das Textformatierungsprogramm `$fat nroff$`. Der `$fat neqn$` Präprozessor liest Files, die Text und mathematische Ausdrücke vermischt enthalten und gibt seinen Output an `$fat nroff$`. Das Programm `$fat neqn$` fügt für die Abschnitte, die von `$fat neqn$` zu behandeln sind, Steuerfolgen in das File ein, die von `$fat nroff$` verarbeitet werden. Es baut wie `$fat nroff$` auf die Fähigkeiten von Seriendruckern auf. Die Anpassung an gerätespezifische Besonderheiten, z.B. Zeilenabstand, Zeichenabstand u.a., kann durch Modifikation der Gerätetabelle erreicht werden.

## 2. Elemente von neqn

~~~~~

2.1. Hervorheben von Gleichungen

~~~~~

Um dem Programm \$fat neqn\$ mitzuteilen, wo ein mathematischer Ausdruck beginnt und endet, wird er durch Zeilen markiert, die mit .EQ und .EN beginnen.

Wenn die Zeilen

```
.EQ
x=y+z
.EN
```

sich in einem Eingabefile befinden, wird die Zeile

im Ausgabefile erscheinen. Die Zeilen .EQ und .EN bleiben unverändert. Sie werden nicht anderweitig durch \$fat neqn\$ verarbeitet. Das bedeutet, daß der Anwender Aufgaben wie Zentrierung, Numerierung selbst realisieren muß. Der einfachste Weg ist die Nutzung des \$fat nroff\$-Makropaketes '-ms', das Zentrierung, Einrückung, Linksausrichtung und Numerierung der Gleichungen erlaubt.

Mit dem '-ms'-Paket werden Gleichungen standardgemäß zentriert. Um eine Gleichung links auszurichten, ist .EQ L anstelle von .EQ zu benutzen. Um eine Einrückung auszuführen, ist .EQ I zu benutzen. Jeder dieser Ausdrücke kann durch eine beliebige Gleichungsnummer gefolgt werden, die am rechten Rand plaziert wird.

Zum Beispiel erzeugt der Eingang

```
.EQ I (2.1a)
x=f(y/2)+y/2
.EN
```

den Ausgang

(2.1a)

Es ist auch eine Kurzformnotierung möglich, so daß die Ausdrücke .EQ und .EN nicht ständig wieder eingegeben werden müssen. (siehe Abschnitt 2.2)

### 2.2. Abkürzungen für in-line Gleichungen

~~~~~

In einer Schrift, die mathematische Ausdrücke enthält, ist es oft notwendig, nicht nur Gleichungen hervorzuheben, in dem sie auf einer gesonderten Zeile ausgegeben werden, sondern auch innerhalb einer Textzeile mathematische Ausdrücke anzugeben.

Obgleich dies durch Einschließen des entsprechenden Teiles mit .EQ und .EN erzeugt werden könnte, ist die kontinuierliche Wiederholung von .EQ und .EN unbequem. Außerdem bewirkt .EQ und .EN bei Benutzung des Makropaketes '-ms' eine Hervorhebung der Gleichung auf einer gesonderten Zeile. Das Programm \$fat neqn\$ liefert eine Abkürzungsmöglichkeit für in-line-Ausdrücke. Man kann zwei Zeichen als Begrenzer definieren, um das linke und rechte Ende eines mathematischen Ausdrucks zu markieren. Der linke und rechte Begrenzer können identisch sein. Um

sowohl den linken als auch den rechten Begrenzer auf das Währungszeichen zu setzen, sind z.B. am Beginn des Dokumentes oder spätestens vor der ersten Benutzung der Begrenzer die drei Zeilen

```
.EQ
delim $$
.EN
einzufügen.
```

Wenn dies erfolgt ist, kann z.B. ein Text wie der folgende in einem Dokument ausgegeben werden.

Eine Determinante läßt sich nach den Elementen einer beliebigen (i -ten) Zeile nach der Formel $D = a_{i1} A_{i1} + a_{i2} A_{i2} + \dots + a_{in} A_{in}$ entwickeln; die A_{ij} sind hierbei die Adjunkten der entsprechenden Elemente.

Leerzeichen und neue Zeilen sind nur in dem Text signifikant, nicht jedoch in dem Gleichungsteil selbst. Mehr als eine Gleichung können in einer einzelnen Eingabezeile vorhanden sein.

Es wird genügend Platz vor und nach einer Zeile gelassen, wenn die Eingabezeile einen Ausdruck enthält, etwa wie $\sum_{i=1}^n x_i$, der nicht mit den umgebenden Zeilen übereinstimmt.

Das Ausschalten der Begrenzer erfolgt durch die Angaben

```
.EQ
delim off
.EN
```

Es ist darauf hinzuweisen, daß geschweifte Klammern, Tilden, Zirkumflexe oder Anführungszeichen nicht als Begrenzer zu verwenden sind.

2.3. Behandlung von Leerzeichen bei der Ein- und Ausgabe

~~~~~

Leerzeichen und neue Zeilen innerhalb eines mathematischen Ausdruckes werden durch neqn vernachlässigt. Folglich ergibt

$x=y+z$

oder

$x = y + z$

oder

$x = y$   
 $+z$

zwischen .EQ und .EN immer den gleichen Ausgang

Man sollte aber trotzdem Leerzeichen und neue Zeilen in den Eingabegleichungen benutzen, um sie lesbar zu machen und um sie leichter editieren zu können.

\$fat Neqn\$ erkennt spezielle Worte in einem Eingabefile, für die spezielle Symbole im Ausgabefile eingefügt werden. Diese speziellen Worte sind im Eingabefile von anderen Worten durch Leerzeichen (Tabulatoren oder neue Zeilen) zu trennen, um als solche erkannt zu werden. Spezielle Worte können auch durch das Einschließen mit Tilden oder Zirkumflex kenntlich gemacht werden.

Um Leerzeichen bei der Ausgabe eines mathematischen Ausdrucks zu erzwingen, kann für jedes gewünschte Leerzeichen eine Tilde "~" im Eingabefile angegeben werden. Der Ausdruck im Eingabefile

x~~y~+~z

ergibt

Tabulatoren können auch benutzt werden, um Teile einer Gleichung zu positionieren.

#### 2.4. Indizes

~~~~~

Hoch- und tiefgestellte Indizes werden mit den Schlüsselworten \$fat "sub"\$ und \$fat "sup"\$ erzeugt. Der Ausdruck

x sup 2 + y sub k

in einem Eingabefile führt zu der Ausgabe

\$fat Neqn\$ beachtet alle vertikalen Bewegungen, die notwendig sind, um den Ausgang ordnungsgemäß zu erzeugen. Die Worte \$fat "sub"\$ und \$fat "sup"\$ müssen von Leerzeichen umgeben sein (x sub2 wird \$x sub2\$ ergeben, anstatt \$x sub 2\$). Weiterhin ist nicht zu vergessen, ein Leerzeichen (oder eine Tilde usw.) zu lassen, um das Ende eines hoch- oder tiefgestellten Index zu markieren. Ein allgemeiner Fehler ist z.B. anzugeben

y = (x sup 2)+1

was

ergibt, anstelle der erwarteten Ausgabe

Doppelt hoch- und tiefgestellte Indizes können auch konstruiert werden. Die Angabe

x sub i sub 1

ergibt im Ausgabefile

Ein tief- und hochgestellter Index der sich auf die gleiche Basis bezieht, wird in der gleichen vertikalen Lage gedruckt, wenn der tiefgestellte Index zuerst angegeben wird. So führt die Angabe

x sub i sup 2

im Eingabefile zu der folgenden Ausgabe:

Normalerweise wird das Ende eines hoch- oder tiefgestellten Index durch ein Leerzeichen (Tabulator oder Tilde) markiert. Soll als Index ein komplexer Ausdruck, der selbst Leerzeichen enthält, gedruckt werden, so können die Klammern { und } benutzt werden, um den Beginn und das Ende des Index zu markieren.
Um

im Ausgabefile zu erhalten, ist folgender Ausdruck

$$e^{\sup{x^2 + y^2}}$$

im Eingabefile anzugeben. Allgemein kann gesagt werden, daß Klammern immer benutzt werden können, um $\$fat\ neqn\$$ mitzuteilen, daß irgendein Ausdruck als eine Einheit zu behandeln ist. Klammern können auch innerhalb von Klammern vorkommen, wenn es erforderlich ist. Sollen geschweifte Klammern gedruckt werden, sind sie in Anführungszeichen einzuschließen (z.B: "{").

2.5. Brüche

~~~~~

Um einen Bruch darzustellen, ist das Schlüsselwort  $\$fat\ "over"\$$  zu benutzen. Der Ausdruck

$$a+b\ over\ 2c = 1$$

ergibt

Der Bruchstrich wird von  $\$fat\ neqn\$$  in richtiger Länge erzeugt und automatisch positioniert. Klammern können zur eindeutigen Darstellung benutzt werden. Befindet sich nur ein  $\$fat\ "sup"\$$  und ein  $\$fat\ "over"\$$  in einem Ausdruck, so gibt  $\$fat\ neqn\$$  das  $\$fat\ "sup"\$$  einen höheren Vorrang als das  $\$fat\ "over"\$$ . So führt der Ausdruck

$$-b\ sup\ 2\ over\ c$$

zu  $\$ -b\ sup\ 2\ over\ c\$$  anstelle von  $\$ -b\ sup\ {2\ over\ c}\ \$$ .

## 2.6. Quadratwurzeln

~~~~~

Um eine Quadratwurzel darzustellen, ist das Schlüsselwort $\$fat\ "sqrt"\$$ zu benutzen. So führt der Ausdruck

$$sqrt\ a+b + 1\ over\ sqrt\ {ax\ sup\ 2 + bx + c}$$

zu der Ausgabe

Quadratwurzeln über großen Ausdrücken sehen unvollkommen aus, weil ein Wurzelzeichen über einem großen Ausdruck schwierig darstellbar ist. So führt die Angabe

$$sqrt\ {a\ sup\ 2\ over\ b\ sup\ 2}$$

im Eingabefile zu der folgenden Ausgabe:

Es ist günstiger, große Quadratwurzeln mit der Potenzzahl 1/2 darzustellen.

wird durch folgenden Ausdruck erhalten:
 $(a^2 / b^2)^{1/2}$.

2.7. Spezielle Zeichen

~~~~~

\$fat Neqn\$ realisiert die Darstellung einiger mathematischer Zeichen.  
 Um z.B. folgenden Ausdruck

im Ausgabefile zu erhalten, ist im Eingabefile einzugeben:

$\sum_{i=0}^{\infty} x^i$

In diesem Ausdruck werden Klammern benutzt, um anzuzeigen, wo der obere Teil \$ i = inf \$ beginnt und endet. Für den unteren Teil, i=0, sind keine Klammern notwendig, da er keine Leerzeichen enthält. Klammern können immer zur eindeutigen Darstellung angegeben werden. Wenn der Ausdruck des \$fat "from"\$ und \$fat "to"\$ Teiles Leerzeichen enthält, ist dieser Ausdruck in Klammern einzuschließen.

Die Angabe des \$fat "from"\$ und \$fat "to"\$ Teiles ist wahlfrei. Wenn beide benutzt werden, sind sie aber in dieser Reihenfolge anzugeben.

Das folgende Beispiel enthält nur einen \$fat "from"\$ Teil, der in Klammern zu setzen ist. Der Ausdruck

$\lim_{n \rightarrow \infty} x^n = 0$

ergibt im Ausgabefile

Neben dem Summenzeichen können in einer Schrift durch Angabe der Schlüsselworte \$fat "int"\$, \$fat "prod"\$ und \$fat "union"\$ das Integral- Produkt- und Unionzeichen erzeugt werden.

Aus

int prod union

werden

\$int\$ \$prod\$ \$union\$ .

### 2.8. Änderungen der Schriftart

~~~~~

\$fat Neqn\$ gibt die Möglichkeit, in einem Text eingestreute mathematische Ausdrücke in den Schriftarten normal (roman-R), kursiv (italic-I) und Breitdruck (bold-B) zu erzeugen. Als Standard erfolgt die Ausgabe in der Schriftart normal. Ein Wechsel der Schriftart kann durch die Schlüsselworte

\$roman "roman"\$ \$italic "italic"\$ \$bold "bold"\$

erreicht werden. Breitdruck ist abhängig von den Fähigkeiten des Ausgabegerätes. Die Zeichen werden auf die doppelte Breite vergrößert.

Die Änderung der Schriftart wirkt nur auf den Textteil, der dem Schlüsselwort unmittelbar folgt. Danach erfolgt die Rückkehr zur normalen Ausgabe. So führt die Angabe

x y

zu der Ausgabe

Soll ein komplizierter Ausdruck in einer bestimmten Schriftart ausgegeben werden, so ist dieser Ausdruck in Klammern zu setzen. Sollen alle mathematischen Ausdrücke in einer bestimmten Schriftart ausgegeben werden, so kann diese Schriftart durch die Anweisungszeilen

```
.EQ
gfont n
.EN
```

global festgelegt werden. Die globalen Angaben der Schriftart in einem Dokument können mehrfach geändert werden, wenn dies erforderlich ist. Für n ist entsprechend der gewünschten Schriftart R, I oder B bzw. 1, 2 oder 3 einzusetzen.

Diese Wirkung kann auch erzielt werden, wenn der Aufruf des Programms `$fat neqn$` mit dem Kommandozeilen-Argument `$fat -fn$` erfolgt, wobei für n die obigen Parameter einzusetzen sind.

Mit Hilfe der Operation `$fat"fat"$` kann ein `$fat Fettdruck$` erzeugt werden. So führt die Angabe

```
fat { x sub i }
```

zu der Ausgabe

```
$fat { x sub i }$ .
```

2.9. Unterscheidungszeichen

~~~~~

Um Sonderzeichen über Buchstaben zu erhalten, gibt es verschiedene Möglichkeiten:

|          |              |
|----------|--------------|
| x dot    | \$x dot\$    |
| x dotdot | \$x dotdot\$ |
| x hat    | \$x hat\$    |
| x tilde  | \$x tilde\$  |
| x vec    | \$x vec\$    |
| x dyad   | \$x dyad\$   |
| x bar    | \$x bar\$    |
| x under  | \$x under\$  |

Die Unterscheidungszeichen werden in der richtigen Höhe gesetzt. `$fat "Bar"$` und `$fat "under"$` werden über die gesamte Länge einer Konstruktion gesetzt, wie in `$x+y+z bar$`; die anderen Zeichen werden zentriert. Es sind auch Kombinationen der Unterscheidungszeichen möglich. Die Angabe

```
y dotdot bar~~f(t) bar
```

führt zur Ausgabe

## 2.10. Apostrophieren von Text

---

Jeder Input, der in Anführungszeichen ("...") eingeschlossen ist, wird unverändert weitergegeben. Dies erlaubt, eine eigene Einfügung von Leerzeichen und Ausrichtung einzuführen, wenn dies benötigt wird.

Anführungszeichen werden auch benutzt, um Klammern und Schlüsselworte für `$fat neqn$` zu drucken.

Die Konstruktion `" "` ist auch dann erforderlich, wenn `$fat neqn$` aus syntaktischen Gründen die Angabe eines Ausdruckes verlangt, obwohl im Output keine Entsprechung hierfür erscheint. Um z.B. `$ " " sup 2 roman He $` zu erzeugen, kann man nicht angeben: `sup 2 roman He`, weil `$fat "sup"$` ein hochgestellter Index ist, der sich auf eine bestimmte Basis bezieht. Aus diesem Grunde ist anzugeben:

```
" " sup 2 roman He.
```

## 2.11. Ausrichten von Gleichungen

---

Oft ist es notwendig, eine Reihe von Gleichungen nach einer bestimmten vertikalen Position, oft nach einem Gleichheitszeichen, auszurichten. Dies wird mit den zwei Operationen `$fat "mark"$` und `$fat "lineup"$` realisiert.

Das Schlüsselwort `$fat "mark"$` kann einmal an jedem Platz in einer Gleichung auftreten. Es hält die horizontale Position fest, vor der es erscheint. Nachfolgende Gleichungen können einmal das Schlüsselwort `$fat "lineup"$` enthalten. Die Stelle, an der `$fat "lineup"$` in der Gleichung auftritt, wird festgehalten, um sie zu dem Platz horizontal auszurichten, der durch das vorhergehende `$fat "mark"$` markiert ist, wenn dies möglich ist. Folglich

kann man z.B. angeben

```
.EQ I
x+y mark = z
.EN
.EQ I
x lineup = 1
.EN
```

um

zu erhalten. Wenn `$fat neqn$` und `'-ms'` benutzt werden, kann `.EQ I` oder `.EQ L` verwendet werden, um eine Einrückung oder Linksausrichtung einer Gleichung zu erzielen.

Es ist zu beachten, daß `$fat "mark"$` den Platzbedarf für nachfolgende Gleichungen nicht berücksichtigen kann. Deshalb werden die Zeilen

```
x mark = 1
x+y lineup = z
```

zu keinem korrekten Ergebnis führen, da nicht genügend Platz für den "x+y - Teil" reserviert wird, da `$fat "mark"$` nur festhält, an welcher Position x sich befindet.

## 2.12. Klammern

~~~~~

Mit den Schlüsselworten `$fat "left"$` und `$fat "right"$` können Klammern dargestellt werden. Als Argument können eckige, geschweifte, runde Klammern oder Längsstriche angegeben werden. `Neqn` erzeugt in jedem Fall Längsstriche als Klammern, die groß genug erzeugt werden, um den einzuklammernden Ausdruck zu bedecken.

Die Angabe

```
left { a over b + 1 right }
  ~~= left ( c over d right )
    + left [ e right ]
```

im Eingabefile wird

im Ausgabefile.

Klammern dieser Art können auch mit den Argumenten `"f" ('floor')` und `"c" ('ceiling')` dargestellt werden.

Die Angabe

```
left floor x over y right floor
  <= left ceiling a over b right ceiling
```

erzeugt

Der `$fat "right"$` Teil ist optional und kann weggelassen werden. Wenn der `$fat "left"$` Teil weggelassen werden soll, ist zu beachten, daß aus programmtechnischen Gründen kein `'right'` ohne ein dazugehöriges `'left'` erzeugt werden kann. Aus diesem Grunde ist z.B. anzugeben:

```
left "... right )
```

Das `left ""` bedeutet, daß nichts ausgegeben wird. Die syntaktischen Regeln werden durch diese Angabe aber nicht verletzt.

2.13. Spalten

~~~~~

`$fat Neqn$` liefert eine Möglichkeit, vertikale Spalten von Ausdrücken zu erzeugen. Zum Beispiel führt die Angabe

```
A~~~ left [ pile { a above b above c }
  ~~~ pile { x above y above z }
 right]
```

zu der Ausgabe

Die Zahl der Elemente in einer Spalte ist nicht begrenzt. Sie werden zentriert in der richtigen Höhe ausgegeben. Das Schlüsselwort `$fat "above"$` wird benutzt, um die Elemente zu trennen. Klammern werden um die ganze Liste gesetzt. Die Elemente einer Spalte können so kompliziert sein, wie sie benötigt werden; sie können sogar weitere Spalten enthalten.

Neben dem Schlüsselwort `$fat "pile"$` gibt es noch die Schlüsselworte `$fat "lpile"$`, `$fat "rpile"$` und `$fat "cpile"$` zur Darstellung von Spalten. Mit `$fat "lpile"$` wird eine Spalte mit links ausgerichteten Elementen erzeugt; `$fat "rpile"$` erzeugt eine Spalte mit rechts ausgerichteten Elementen und `$fat "cpile"$` eine Spalte mit zentrierten Elementen. Die vertikalen Abstände zwischen den Elementen sind etwas größer für `$fat l-$`, `$fat r-$` und `$fat "cpile"$`,

als es für Spalten, die mit `$fat "pile"$` erzeugt sind, der Fall ist.  
Die Angabe

```
roman sign (x)~~~
 left {
 lpile { 1 above 0 above -1 }
 ~~lpile
 { if~x>0 above if~x=0 above if~x<0 }
```

führt zu der Ausgabe

Es sei darauf hingewiesen, daß in dem Ausdruck eine linke Klammer verwendet wird, ohne eine entsprechende rechte Klammer anzugeben.

## 2.14. Matrizen

~~~~~

Es ist auch möglich, mit Hilfe von `$fat neqn$` Matrizen zu erzeugen.  
Um z.B. ein Feld wie

zu erzeugen, ist im Eingabefile anzugeben:

```
matrix {
 ccol {x sub i above y sub i}
 ccol {x sup 2 above y sup 2}
}
```

Dies erzeugt eine Matrix mit zwei zentrierten Spalten. Die Elemente der einzelnen Spalten einer Matrix werden wie für die Ausgabe einer separaten Spalte gelistet. Jedes Element wird durch das Wort `$fat "above"$` getrennt. Man kann auch `$fat "lcol"$` oder `$fat "rcol"$` benutzen, um Spalten nach links oder rechts auszurichten. Jede Spalte kann separat ausgerichtet werden, und es können so viele Spalten vorhanden sein, wie es vom Anwender gewünscht wird.

Der Grund für die Benutzung des Schlüsselwortes `$fat "matrix"$` zur Darstellung einer Matrix anstelle des Schlüsselwortes `$fat "pile"$` zur Darstellung von zwei nebeneinander liegenden Spalten ist der, daß bei Spaltenelementen unterschiedlicher Größe die einzelnen Elemente nicht ordnungsgemäß ausgerichtet werden. Nur bei Benutzung des Schlüsselwortes `$fat "matrix"$` wird die gesamte Struktur der Matrix beachtet, bevor entschieden wird, wieviel Leerzeichen und -zeilen zu benutzen sind, um die einzelnen Spalten ordnungsgemäß auszurichten.

Bei der Benutzung des Schlüsselwortes `$fat "matrix"$` ist unbedingt zu beachten, daß jede Spalte die gleiche Zahl von Elementen enthält.

## 2.15. Definitionen

~~~~~

`$fat Neqn$` bietet die Möglichkeit, einer häufig benutzten Zeichenkette einen Namen zu geben, und danach nur den Namen zu benutzen, anstelle der gesamten Kette. Wenn z.B. die Folge

```
x sub i sub 1 + y sub i sub 1
```

wiederholt in einer Schrift erscheint, kann man sich das ständige Wiedereingeben sparen, durch eine Definition wie die folgende:

```
define xy 'x sub i sub 1 + y sub i sub 1'
```

Dies definiert `xy` als eine Abkürzung für die Zeichen, die zwischen den Apostrophs in der Definition erscheinen. Es kann jedes Zeichen anstelle des Apostrophs benutzt werden, um die Enden der Definition zu markieren, solange das

Zeichen nicht innerhalb der Definition erscheint. Nun kann xy wie folgt benutzt werden:

```
.EQ
f(n) = xy ...
.EN
```

Bei jedem Auftreten wird xy durch die Zeichenkette ersetzt, die es definiert. Es ist zu beachten, daß Leerzeichen oder ihre äquivalente um die Abkürzung einzufügen sind, wenn sie aktuell benutzt wird, so daß \$fat neqn\$ sie als ein spezielles Zeichen identifizieren kann.

In Definitionen können vorherige Definitionen benutzt werden, wie in dem folgenden Beispiel gezeigt wird.

```
.EQ
define xi 'x sub i'
define xil 'xi sub 1'
.EN
```

Der zu definierende Ausdruck darf diese Definition aber nicht selbst enthalten. Ein Fehler ist es, anzugeben:

```
define X 'roman X'.
```

Wenn man dagegen angibt

```
define X 'roman "X"'
```

führt dies zu keinem Fehler, da die Anführungszeichen ein Ersetzen des zweiten X verhindern.

Mit der \$fat "define"\$-Anweisung können auch Schlüsselworte von \$fat neqn\$ umdefiniert werden. Man kann / als \$fat "over"\$ definieren durch die Angabe

```
define / 'over'
```

und kann es durch die Angabe

```
define over '/'
```

zurückdefinieren.

## 2.16. Horizontale und vertikale Bewegungen

~~~~~

Vom Programm \$fat neqn\$ werden alle erforderlichen Positionierungen bei der Ausgabe mathematischer Ausdrücke vorgenommen. Wünscht der Anwender selbst Positionierungen vorzunehmen, stehen ihm hierfür die Schlüsselworte \$fat "fwd"\$ und \$fat "back"\$ für horizontale Bewegungen, sowie \$fat "up"\$ und \$fat "down"\$ für vertikale Bewegungen zur Verfügung.

Die Größe der gewünschten Bewegung wird durch den Parameter n bestimmt. Der Parameter n gibt allgemein an, wieviel Hundertstel der Weite des Buchstaben `m' zu bewegen ist. Die tatsächliche Bewegung hängt jedoch von der Geräteauflösung ab. (Die kleinsten Bewegungseinheiten der Drucker SD1152/57 betragen 1 Zeichen vor- oder rückwärts, bzw. eine halbe Zeile auf- oder abwärts.)

So führt die Angabe

```
.EQ
abcd $down 50 efgh$
.EN
```

zu der Ausgabe

```
abcd $down 50 efgh$.
```

## 2.17. Vorränge der Schlüsselworte

~~~~~

Wenn vom Anwender keine Klammern benutzt werden, wird das Programm \$fat neqn\$ die Operationen in der Reihenfolge ausführen, wie sie in folgender Liste aufgeführt sind.

dyad vec under bar tilde hat dot dotdot  
 fwd back up down  
 fat roman italic bold  
 sub sup sqrt over  
 from to

Folgende Operationen gruppieren links:

over sqrt left right

Alle anderen gruppieren rechts.

Zahlen, Klammern, Interpunktionszeichen und die folgenden mathematischen Worte werden in der Schriftart `roman' ausgegeben.

sin cos tan sinh cosh tanh arc  
 max min lim log ln exp  
 Re Im and if for det

Folgende Zeichenfolgen werden erkannt und in der folgenden Form übersetzt:

|         |             |
|---------|-------------|
| >=      | \$>=\$      |
| <=      | \$<=\$      |
| ==      | \$==\$      |
| !=      | \$!=\$      |
| + -     | \$+-\$      |
| ->      | \$->\$      |
| <-      | \$<-\$      |
| <<      | \$<<\$      |
| >>      | \$>>\$      |
| inf     | \$inf\$     |
| half    | \$half\$    |
| prime   | \$prime\$   |
| approx  | \$approx\$  |
| nothing | \$nothing\$ |
| cdot    | \$cdot\$    |
| times   | \$times\$   |
| del     | \$del\$     |
| grad    | \$grad\$    |
| sum     | \$sum\$     |
| int     | \$int\$     |
| prod    | \$prod\$    |
| union   | \$union\$   |

### 3. Fehlermeldungen

~~~~~

Wenn ein Fehler in einer Gleichung gemacht wird, wie das Auslassen einer Klammer, die Angabe einer unnötigen Klammer, oder die Angabe eines \$fat "sup"\$ ohne vorhergehenden Parameter, wird vom Programm \$fat neqn\$ folgende Mitteilung ausgegeben:

```
 syntax error between lines x and y, file z
```

wobei x und y die Zeilen angeben, zwischen denen der Fehler auftritt und z der Name des in Frage kommenden Files ist.

Es werden auch einige Fehlermitteilungen ausgegeben, die sich selbst erklären, z.B. wenn ein Apostroph weggelassen wird, oder wenn das als Parameter angegebene File nicht existiert.

Wenn man eine Schrift prüfen will, bevor sie gedruckt wird, kann man folgendes Kommando eingeben:

```
 neqn files >/dev/null
```

In diesem Fall wird kein Ausgang erzeugt, aber die Fehlermitteilungen werden ausgegeben.

Wenn man z.B. das Währungszeichen als Begrenzer benutzt, kann es leicht vorkommen, daß ein Zeichen vergessen wird. Dies kann sehr viele Folgefehler verursachen. Mit Hilfe des Programms \$fat checkeq\$ kann man auf falsch platzierte oder fehlende Begrenzerzeichen und ähnliche Fehler prüfen.

In-line-Gleichungen können nur so groß sein, wie ein interner Puffer im Programm \$fat nroff\$. Wenn man die Mitteilung "word overflow" erhält, ist diese Grenze überschritten. Wenn man diese Gleichung auf einer gesonderten Zeile druckt, wird diese Mitteilung gewöhnlich nicht erscheinen. Die Mitteilung "line overflow" zeigt an, daß man einen noch größeren Puffer überschritten hat. In diesem Fall soll die Gleichung in zwei separate Teile aufgeteilt werden.

Das Programm \$fat neqn\$ unterteilt Gleichungen nicht selbständig. Lange Gleichungen sind vom Anwender selbst aufzuteilen, indem jeder Teil durch eine separate EQ...EN-Folge markiert wird. Das Programm \$fat neqn\$ warnt nicht über Gleichungen, die zu lang sind, um auf eine Zeile zu passen.

#### 4. Benutzung im System MUTOS

~~~~~

Um ein Dokument, das mathematische Ausdrücke enthält, unter Benutzung des Systems MUTOS zu drucken, ist folgende Kommandoeingabe erforderlich.

```
neqn files | nroff
```

Options des Kommandos \$fat nroff\$ können nach dem \$fat nroff\$-Teil des Kommandos angegeben werden, z.B.

```
neqn file | nroff -ms
```

Um einen speziellen Drucker als Ausgabegerät zu spezifizieren, ist anzugeben

```
neqn files | nroff -Tx ,
```

wobei x den verwendeten Druckertyp angibt. Das Programm \$fat neqn\$ kann auch mit dem Programm \$fat tbl\$ zum Setzen von Tabellen, die mathematische Ausdrücke enthalten, verwendet werden. \$fat Tbl\$ ist in diesem Fall vor \$fat neqn\$ anzuwenden.

```
tbl files | neqn | nroff
```