Fassung vom 15.10.1996

Anleitung zu mit Edmac kompatiblen Tabellen

von Nora Gädeke und Herbert Breger

Vorbemerkung: Der folgende Text ist ein Auszug aus der am Leibniz-Archiv Hannover (Waterloostr. 8, 30169 Hannover, Tel.: 0511 1267 327 oder 0511 1267 322) verwendeten Anleitung für TEX und Edmac.

Allgemeines

Als Tabelle bezeichnen wir im folgenden jedes Gebilde, das zum einen mehrzeilig ist und zum anderen aus mehreren Spalten besteht, deren einzelne Zeilen in irgendeiner Weise bündig oder symmetrisch aufeinander ausgerichtet sind. In einer Tabelle können sowohl Text als auch Formeln vorkommen. TEX bietet sehr schöne Möglichkeiten zur Tabellenherstellung, die aber für unsere Zwecke nach gegenwärtigem Stand nicht in Frage kommen, weil die Zeilenzählung aussetzt und die Tabelle nicht zentriert ausgegeben wird (Probleme mit EDMAC bei `\vbox`). Diese Probleme werden durch die selbsterstellten Macros für Tabellen umgangen.

Ausgerichtete Gleichungen (Tabellen im Mathematikmodus)

Zunächst ist grundsätzlich zu sagen, daß alle mathematischen Tabellenmacros alle Elemente automatisch im displaystyle bringen.

Nun zu den einzelnen Macros:

Die bei weitem angenehmste Möglichkeit, eine Tabelle zu erstellen, bieten die Macros `\rtab`, `\ltab` bzw. `\ztab`, bei Wunsch in Verbindung mit `\abstand`. Sie lassen sich allerdings nur bei einheitlicher Ausrichtung aller Tabellenelemente verwenden: mit `\rtab` erhält man eine rechtsbündige, mit `\ltab` eine linksbündige Tabelle und mit `\ztab` eine Tabelle, in der die Elemente zentriert sind. Die Macros `\rtab`, `\ltab` und `\ztab` sind auf maximal sechzig Spalten ausgerichtet (Erweiterung wäre sehr leicht möglich, ist aber

---

10 Dies beginnt also bereits mit quadratischen Gleichungen, deren Koeffizienten zweizeilig untereinander angeordnet sind.

vermutlich kaum erforderlich). Die Anzahl der Zeilen ist unbegrenzt. Das Problem einer Tabelle, in der nur einzelne Elemente zentriert sein sollen, kann man mit `\phantom` lösen. Da die Unterscheidung rechtsbündig-linksbündig innerhalb einer Tabelle hier entfällt, sind auch keine zwei verschiedenen Positionszeichen nötig (wie es bei `\atab`, vgl. Seite 10, der Fall ist): nach jedem Tabellenelement schreibt man ein & (außer nach dem letzten Element in einer Zeile). Nach jedem & sollte man ein blank lassen. (Fehlermeldungen wie "Anzahl der Tabulatoren stimmt nicht überein mit der Anzahl der Tabulatoren in der vorangehenden Zeile" können auf ein solches fehlendes blank zurückzuführen sein.) Abgeschlossen wird eine jede Zeile (außer der letzten) mit einem `!`. Die gesamte Tabelle wird in geschweifte Klammern gesetzt. Nun ein Beispiel: Die Eingabe

```
    \rtab{ 2xyz & +& \text{a}\dfootnote{ Erläuterung}/& - &{\mu\over x^2}
& +& r^2& = & 0 !
    4b & -& bgh& + & ac & - & z & = & 0 !
    a & +& 4c & - & r & + & \sqrt{vrft} & = & 0 !
    & & & - & p & + & v^4 & = & 0 }
```

ergibt

$$\begin{array}{rcrcrcrcl} 2xyz & + & a & - & \dfrac{\mu}{x^2} & + & r^2 & = & 0 \\ 4b & - & bgh & + & ac & - & z & = & 0 \\ a & + & 4c & - & r & + & \sqrt{vrft} & = & 0 \\ & & & - & p & + & v^4 & = & 0 \end{array}$$

Falls der Spaltenabstand zu klein oder zu groß sein sollte, kann man ihn mittels `\abstand` variieren. Vorgegeben ist zunächst ein Abstand von 10pt. Will man ihn z.B. auf 15pt verändern, so setzt man vor Beginn der Tabelle `\abstand=15pt`. Dieser Wert ist dann nur für diese Tabelle gültig, danach gilt automatisch wieder `\abstand=10pt`.

Wie man sieht, erscheint alles automatisch im displaystyle. Es ist nicht erforderlich, vorher in den Mathematikmodus zu gehen, da dies automatisch geschieht. Falls man allerdings bei Beginn der Tabelle schon im Mathematikmodus ist, muß man diesen erst verlassen. Will man innerhalb der Tabelle Text schreiben, so kann man dies durch `{\rm Textwort}` tun. Hat man allerdings mehrere Textworte, so würde TEX – da man sich ja im Mathematikmodus befindet – die Abstände zwischen den Worten verschwinden lassen. Man sollte daher `{\rm Textwort ~ Textwort }` eingeben.

Wichtig: Bei `\rtab`, `\ltab`, `\ztab`, `\rtabtext` und `\ltabtext` darf man nach der Tabelle keine Leerzeile verwenden, sondern muß `\pend\pstart` schreiben, weil sonst

---

17 a: Erläuterung

footnotes und labels verschwinden können.

Sollte man (was unwahrscheinlich ist) innerhalb der Tabelle ein ! schreiben wollen, so kann man dies, indem man das Ausrufezeichen in eine geschweifte Klammer setzt: {!}. Dann wird das Zeichen von TEX nicht als Zeilenende gelesen.

Hat man sich bei `\rtab`,`\ltab` oder `\ztab` bei den Tabulatoren verzählt, so erhält man eine Fehlermeldung. Sie weist auf die fehlende Übereinstimmung in der Zahl der Tabulatoren hin. Wenn man beispielsweise in der ersten Zeile einen Tabulator zuviel oder zuwenig gesetzt hat, erhält man in der zweiten Zeile einen Hinweis auf die fehlende Übereinstimmung in der Anzahl der Tabulatoren. Wenn man in der zweiten Zeile einen Tabulator zuviel oder zuwenig gesetzt hat, so erhält man eine Fehlermeldung für die zweite und die dritte Zeile (weil jeweils die Anzahl der Tabulatoren nicht mit der vorangehenden Zeile übereinstimmt). Diese Fehlermeldung hat jedoch nur Kontrollfunktion: denn man kann durchaus Tabulatoren (wenn z.B. in einigen Zeilen die letzten Spalten leer bleiben sollen) weglassen, ohne die Tabelle in Gefahr zu bringen. Man bekommt dann per Fehlermeldung den Hinweis auf die fehlende Übereinstimmung in der Zahl der Tabulatoren, aber das kann man ignorieren, wenn man sich sicher ist, keinen Fehler gemacht zu haben.

In der praktischen Ausführung ist `\rtab` etc. meistens `\atab` (vgl. Seite 10) vorzuziehen – sofern die Orientierung der Elemente nicht wechselt. In einem Fall ist aber anzuraten, auf das kompliziertere Macro zurückzugreifen: wenn die Tabelle unterbrochen und nach einigen Zeilen normalen Textes wieder fortgeführt wird. Hier ist die – in diesem Fall immer wieder neu (und vor allem immer wieder gleich) zu setzende – Musterzeile wichtig.

Formeln oder Text über die Spaltenbreite hinaus: Wenn man in einer Tabelle (die mit `\rtab`,`\ltab`, `\ztab`,`\rtabtext` oder `\ltabtext` gemacht ist) ausnahmsweise einmal über die Spaltenbreite hinausschreiben will, so kann man dies mittels `\breiter` (entspricht in etwa `\multispan`). Beispiel: Die Eingabe

```
\ltab{ a& s & huiz & K & H & O & z !
& & \breiter{a+b+c+d+e} & & & & !
a& s & h& K & H & O & z !
a& s & h& K & H & O & z }
```

ergibt

$$\begin{array}{lllllll}
a & s & huiz & K & H & O & z \\
 & & \multicolumn{5}{l}{a+b+c+d+e} \\
a & s & h & K & H & O & z \\
a & s & h & K & H & O & z
\end{array}$$

Nur bei `\ltab` wird das mit `\breiter` Eingegebene auch wirklich bündig ausgegeben. Bei `\rtab` könnte man sich zum Beispiel dadurch helfen, daß man in der Spalte davor den gewünschten Ausdruck eingibt und ihn mittels `\phantom` so weit verschiebt wie erforderlich, also: `\phantom{...}\breiter{...}`. Das nach `\breiter` Folgende wird automatisch im Mathematikmodus und displaystyle gesetzt. Will man Text schreiben, der sich über mehrere Spalten erstreckt, so verwendet man `\breitertext`.

Für die Tabellenmacros `\rtab`, `\ltab` und `\ztab` gibt es die Macros `\davor` und `\danach`, **um links vor dem Beginn der Tabelle bzw. rechts nach dem Ende der Tabelle schreiben zu können**. In dem folgenden Beispiel möchte man vor der Tabelle "Sit" und nach der Tabelle "Erit" schreiben. Da die richtige Position von "Sit" von der Breite des ersten Tabellenelements (und die richtige Position von "Erit" von der Breite des letzten Tabellenelements) abhängt, müssen diese Tabellenelemente mitgeklammert werden. Die Eingabe

```
\lineskip=5pt
\pstart
\abstand=10pt
\rtab
{\davor{Sit}{A}& = & 1\over1 & + & 1\over2 & + & 1\over3 !
            B & = & 1\over1 & + & 1\over3 & + & 1\over6 !
            C & = & 1\over1 & + & 1\over4 & + & 1\over10 !
            D & = & 1\over1 & + & 1\over5 & + &
                                   \danach{{1\over15}}{Erit}}
```

ergibt:

$$\begin{array}{lrcccccccl}
\text{Sit} & A & = & \frac{1}{1} & + & \frac{1}{2} & + & \frac{1}{3} & & \\
 & B & = & \frac{1}{1} & + & \frac{1}{3} & + & \frac{1}{6} & & \\
 & C & = & \frac{1}{1} & + & \frac{1}{4} & + & \frac{1}{10} & & \\
 & D & = & \frac{1}{1} & + & \frac{1}{5} & + & \frac{1}{15} & & \text{Erit}
\end{array}$$

Tabellen im Text

Die bisher erläuterten Tabellenbefehle `\rtab`,`\ltab`, `\ztab` sowie das später zu erörternde `\atab` sind auf den Gebrauch für Mathematik zugeschnitten: Sie sorgen dafür, daß alle Tabellenelemente automatisch im displaystyle erscheinen.

Gelegentlich braucht man aber auch im nicht–mathematischen Text eine Tabelle. Dafür stehen die Macros `\ltabtext` und `\rtabtext` zusammen mit `\abstand` zur Verfügung. Sie sind in jeder Hinsicht den Macros `\ltab` und `\rtab` völlig gleich, mit dem einzigen Unterschied, daß die Tabellenelemente nicht automatisch im Mathematikmodus stehen. Will man über die Spaltenbreite hinaus schreiben, so geschieht dies mittels `\breitertext` (vgl. S. 3)

Tabellen im kritischen Apparat

Die Tabellenmacros sind auch im kritischen Apparat verwendbar, **sofern man Edmac 2.3 verwendet. Bei Verwendung von Edmac 3.16 können Probleme auftreten, die bis jetzt noch nicht untersucht worden sind.**

Man muß im footnote-Format (also in `\Afootfmt`, `\Bfootfmt` bzw. in `\parafootfmt` usw.) `\def\variab{\hfill\penalty-9999}` hinzufügen. Nach dem Bezugswort wird dann mit der Tabelle in einer neuen Zeile begonnen.

Klammern, Striche und Pünktchen in Tabellen

Das Folgende bezieht sich auf `\rtab`, `\ltab` und `\ztab` sowie `\rtabtext` und `\ltabtext`.

Waagerechte Striche: Dafür kann man das Macro `\wastrich` benutzen. Ein waagerechter Strich, der die erste und zweite Spalte umfassen soll, wird in der ersten Spalte eingegeben, und zwar gibt man die Nummern der Spalten mit einem backslash ein und setzt danach einen |. Die Eingabe ist also `\wastrich\eins\zwei|`. Ein Beispiel wird zusammen mit den waagerechten Klammern angegeben.

Falls man direkt in Zahlen angeben möchte, wie lang der Strich sein soll, so schreibt man `\klamdimen=45pt\wastrich\klamdimen|` und erhält einen Strich von 45 pt.

Waagerechte Klammern: Eine nach oben offene Klammer erhält man mit `\waklam`, eine nach unten offene Klammer mit `\Waklam`. Eine waagerechte Klammer, die von der vierten bis zur sechsten Spalte reichen soll, wird in der vierten Spalte eingegeben, und zwar gibt man die Nummern der Spalten mit einem Backslash ein und setzt danach einen |. Die Eingabe ist also: `\waklam\vier\fuenf\sechs|`. (Man beachte, daß man hier "fuenf" statt "fünf" schreiben muß: Ein Umlaut ist nämlich ein verkappter TEX-Befehl, deshalb kann im TEX-Befehl `\fuenf` kein Umlaut und damit ein anderer TEX-Befehl vorkommen).

Die waagerechten Klammern müssen eine gewisse Mindestlänge haben, sonst mißglücken sie. Sollte eine Spalte nicht breit genug sein (weil sie etwa nur einen Buchstaben breit ist), so kann man einfach ein oder zwei geschützte Spatien eingeben, um die erforderliche Mindestbreite der Spalte zu bekommen.

Falls man direkt in Zahlen angeben möchte, wie lang die Klammer sein soll, so schreibt man `\klamdimen=30pt\waklam\klamdimen|` und erhält eine Klammer von 30 pt.

Nun ein Beispiel. Die Eingabe

```
\rtab{\wastrich\eins\zwei| & & fd & h & qwertzuiohg !
v & wpotzug & x & y & vb !
g & nnnn & \waklam\drei\vier\fuenf| & & !
\Waklam\eins\zwei\drei| & & & pq & dgh !
k & & l & co & ghweropjklmnbvcxys }
```

ergibt

		fd	*h*	*qwertzuiohg*
v	*wpotzug*	*x*	*y*	*vb*
g	*nnnn*			
			pq	*dgh*
k		*l*	*co*	*ghweropjklmnbvcxys*

Waagerechte Pünktchen: Man benutzt das Macro `\wapunkte`. Die Eingabe ist ebenso wie bei den waagerechten Strichen und Klammern. Wenn die waagerechten Pünktchen die erste und zweite Spalte umfassen sollen, wird das Macro in der ersten Spalte eingegeben, und zwar gibt man die Nummern der Spalten mit einem Backslash ein und setzt danach einen |. Die Eingabe ist also `\wapunkte\eins\zwei|`. Ein Beispiel: Die Eingabe

```
\rtab{ a & b & C & d & !
\wapunkte\eins\zwei| & & & y & !
m & n & o & p & !
k & & l & cvb & }
```

ergibt

$$\begin{array}{cccc} a & b & C & d \\ \multicolumn{2}{l}{\cdots\cdots} & & y \\ m & n & o & p \\ k & & l & cvb \end{array}$$

Falls man direkt in Zahlen angeben möchte, wie lang die Pünktchen sein sollen, so schreibt man `\klamdimen=20pt\wapunkte\klamdimen|` und erhält Punkte auf einer Länge von 20 pt.

Senkrechte Striche: Der Befehl `\sestrich` erlaubt es, in Tabellen senkrechte Striche zu ziehen. Man gibt `\sestrich` dort ein, wo der Strich sein unteres Ende haben soll. Die Eingabe `\sestrich{45}` erzeugt dann einen senkrechten Strich von 45 pt Höhe.

Senkrechte Klammern: `\seklam` ergibt eine nach links offene Klammer, `\Seklam` ergibt eine nach rechts offene Klammer. Hinter dem Macro gibt man in geschweiften Klammern die Höhe der (Halb-)Klammer in Punkten ein. Man gibt das Macro immer in der Zeile ein, in der die senkrechte Klammer enden soll (also in der untersten Zeile). Beispiel: Die Eingabe

```
\rtab{ a & b & C & d & !
v & w & x & y & !
m & n & o & p & !
k & & l & cvb & \seklam{21}}
```

ergibt

$$\begin{array}{l} \begin{array}{cccc} a & b & C & d \end{array} \\ \left.\begin{array}{cccc} v & w & x & y \\ m & n & o & p \\ k & & l & cvb \end{array}\right\} \end{array}$$

Wenn man einen senkrechten Strich und eine senkrechte Klammer eingeben will, die gleich hoch sind, so muß man verschiedene Zahlenwerte verwenden (der Wert hinter `\seklam` gibt nämlich nur ungefähr die Höhe der Halb-Klammer an). Genauer gesagt: `\sestrich{36}` und `\seklam{21}` sind gleich hoch, ebenso `\sestrich{48}` und

`\seklam{28}`.

Manchmal möchte man neben die senkrechte Klammer noch etwas schreiben, was genau in der richtigen Höhe stehen soll. Wenn die senkrechte Klammer aber über vier, sechs, acht oder zehn Zeilen geht, dann ist die richtige Höhe zwischen den regulären Zeilen. Für diesen Zweck kann man mit den Macros `\seklamgl` und `\Seklamgl` arbeiten. Man gibt hinter `\seklamgl` (in geschweiften Klammern) die Höhe der Klammer in Punkten an, danach in weiteren geschweiften Klammern den weiteren Text. Bei `\Seklamgl` ist die Reihenfolge anders: Man gibt zuerst in geschweiften Klammern an, was neben der Klammerspitze stehen soll, danach in weiteren geschweiften Klammern die Höhe in Punkten. Beispiel: Die Eingabe

```
\rtab{ a & b & C & d & !
v & w & x & y & !
m & n & o & p & !
k & & l & cvb & \seklamgl{29}{= {1\over3}}}
```

ergibt

$$\left.\begin{array}{cccc} a & b & C & d \\ v & w & x & y \\ m & n & o & p \\ k & & l & cvb \end{array}\right\} = \frac{1}{3}$$

Da der Bruch $\frac{1}{3}$ in der letzten Zeile eingegeben wurde, steht er für TeX in Zeile 19. Normalerweise ist das egal, aber wenn man eine Erläuterung dazu machen will, so muß man mit `\label` und `\linenum` arbeiten, um die richtige Zeilennummer zu bekommen. Man gibt in der Zeile, in der die senkrechte Klammer beginnt, ein label ein. Beispiel: Die Eingabe:

```
\rtab{ ga &b & C & d & \label{etwas} !
v &w & x & y & !
m & n & o & p & !
k & & l & cvb & \seklamgl{29}{=
\text{{1\over3}}\linenum{\xpageref{etwas}|\xlineref{etwas}||||}
\lemma{$1\over3$}\dfootnote{Erläuterung}/}
```

ergibt

$$\left.\begin{array}{cccc} ga & b & C & d \\ v & w & x & y \\ m & n & o & p \\ k & & l & cvb \end{array}\right\} = \frac{1}{3}$$

Senkrechte Pünktchen: Mit dem Macro `\sepunkte` kann man senkrechte Punkte erzeugen. Man gibt hinter dem Befehl an, wie hoch die senkrechten Punkte sein sollen. Beispiel: Durch die Eingabe

```
\rtab{ a & b & C & d & !
v & w & x & y & !
m & n & o & p & !
k & & l & cvb & \sepunkte{50}}
```

erhält man eine Linie senkrechter Punkte, die insgesamt 50pt hoch ist:

$$\begin{array}{cccc} a & b & C & d \\ v & w & x & y \\ m & n & o & p \\ k & & l & cvb \end{array} \vdots$$

Eckige Klammern: Mit `\waklamec` kann man eine eckige waagerechte Klammer erzeugen. Die Eingabe ist genau wie die Eingabe von `\waklam` oder von `\wastrich`. Setzt man eine solche Klammer in einer Tabelle, so erscheint sie in der Mitte der Zeile. Beispiel: Die Eingabe

```
\rtab{ axcvgh & brtzj & c !
\waklamec\eins\zwei| & & g !
f & t & zu }
```

ergibt

$$\begin{array}{ccc} axcvgh & brtzj & c \\ \multicolumn{2}{c}{\llcorner\!\!\rule{4cm}{0.4pt}\!\!\lrcorner} & g \\ f & t & zu \end{array}$$

Wenn man eine waagerechte eckige Klammer will, die etwas tiefer als die Zeile sitzt, so verwendet man `\waklamect`. Beispiel: Die Eingabe

```
\lineskip=5pt
\lineskiplimit=2pt
\pstart
\rtab{ \waklamect\eins\zwei| axcvgh & brtzj & c !
```

1–4 $\frac{1}{3}$: Erläuterung

```
rfklo & ftz & g !
f & t & zu }
```

erzeugt

axcvgh	*brtzj*	*c*
rfklo	*ftz*	*g*
f	*t*	*zu*

Statt die Länge der Klammer in Spalten anzugeben, kann man auch mit `\klamdimen` arbeiten und die Länge in Punkten angeben. Die Eingabe

```
\rtab{ \klamdimen=32pt\waklamect\klamdimen| axcvgh & brtzj & c !
rfkop & ftz & g !
f & t & zu }
```

erzeugt

axcvgh	*brtzj*	*c*
rfkop	*ftz*	*g*
f	*t*	*zu*

Ein weiteres Tabellenmacro

Mitunter kommt es vor, daß innerhalb einer Tabelle **einzelne Spalten rechtsbündig, andere aber linksbündig** ausgegeben werden sollen. Für solche Zwecke ist `\atab` brauchbar.

Das Tabellenmacro `\atab` zusammen mit `\atabell` ist für eine variable Zahl von Spalten geeignet, höchstens jedoch für sechzig Spalten (also 120 Positionen). Die Anzahl der Zeilen ist unbegrenzt. Die Positionen erscheinen abwechselnd linksbündig und rechtsbündig, daher der Name. Durch Freilassen von Positionen kann man damit auch eine Tabelle schreiben, in der etwa die erste bis dritte Spalte linksbündig und die vierte bis siebente Spalte rechtsbündig ist.

Zunächst gibt man mit `\atabell` in der Musterzeile die Breite der Tabellenspalten (nicht Positionen!) in mm vor. Hinter jede Zahl (mit Ausnahme der letzten) wird ein & gesetzt, hinter die letzte Zahl lediglich ein ! . Danach gibt man hinter `\atab` die eigentliche Tabelle ein. Man beginnt – nach dem ersten, linksbündigen oder neutralen (Vorzeichen z.B.) Element – mit einem | und setzt dann jeweils abwechselnd ein & und einen |. Jede Spalte (außer der letzten) wird also mit einem & beendet, die Zwischenposition mit einem |. Nach dem letzten | in einer Zeile kommt statt des & ein ! . Die vor einem |

stehenden Elemente werden linksbündig ausgegeben, die vor einem & stehenden Elemente rechtsbündig. Die gesamte Tabelle wird in geschweifte Klammern gesetzt.

Will man Fußnoten, Varianten oder Erläuterungen aus einer mit `\atab` gemachten Tabelle heraus setzen, so darf der Bezugsausdruck, d.h. der unmittelbar nach `\text` in der Klammer stehende Ausdruck, keinen | und kein & enthalten. Ferner muß man den Bezugsausdruck, in `$` (und eventuell `\displaystyle`)... `$` eingeschlossen, in `\lemma` wiederholen. Bei Erläuterungen, Varianten und Fußnoten aus einer Tabelle heraus ist zu beachten, daß die von `\text` geklammerten Worte bzw. Formelausdrücke keinen | und kein & enthalten dürfen. Varianten dürfen aber natürlich trotzdem in senkrechte Striche eingeschlossen werden. Hier ein Beispiel: Die Eingabe

```
\atabell 4 & 19& 8!
\atab{ | a & - | {b \over x^2}&=|0 !
|\text{d^2}\lemma{$d^2$}\dfootnote{Erläuterung}/&+ |
\sqrt{abc^2xy}&=|0 !
| r&-|p&=|0 }
```

ergibt

$$\begin{array}{rrrl} a & - & \dfrac{b}{x^2} = & 0 \\ d^2 & + & \sqrt{abc^2xy} = & 0 \\ r & - & p = & 0 \end{array}$$

Wie man sieht, erscheinen die Elemente automatisch im displaystyle. Am Ende einer mit `\atab` gemachten Tabelle sollte man kein `\pend` eingeben, sondern einfach weiterschreiben, da sich sonst eine mitgezählte leere Zeile ergeben kann. Wenn die erste Position in allen Zeilen nicht besetzt ist, wenn also kein Vorzeichen in der ersten Spalte steht, dann entsteht natürlich der Eindruck, daß die Tabelle nicht zentriert sei, weil der Leerraum links mitberechnet, aber vom Betrachter der Tabelle nicht wahrgenommen wird. Man kann sich in einem solchen Falle helfen, indem man die Breite der ersten Spalte sehr klein wählt.

Wenn man in einer Zeile die letzte Position oder die letzte(n) Spalte(n) nicht mehr braucht, muß man beim Macro `\atab` trotzdem die Tabulatoren hinschreiben.

---

18 d^2: Erläuterung