

# 5 Tabellen

Tabellen sind zweidimensionale Anordnungen von Informationen. Wir betrachten zwei Arten der Spezifikation von Tabellen: Bei der Tabellenbeschreibung mit Hilfe von Tabulatoren werden horizontale Positionen auf der Seite ausgezeichnet, die später angesteuert werden können. Bei der logischen Tabellenbeschreibung werden die Spalten und Zeilen der Tabelle einzeln beschrieben und das Textsystem bestimmt selbständig die horizontale und vertikale Ausdehnung der Tabellenkomponenten. Diese beiden Beschreibungsarten haben unterschiedliche Vorzüge und Hauptanwendungsgebiete.

## 5.1 Tabellen mit Tabulatoren

Bei Schreibmaschinen ist es im allgemeinen möglich, im Inneren einer Zeile eine Reihe von Tabulatorpositionen zu setzen. Beim Druck auf eine spezielle Tabulatortaste bewegt sich die aktuelle Schreibposition nach rechts bis zum nächsten Tabulatorstop. Autoren, die auf diese Art eine Tabelle eingeben wollen, müssen am Anfang der Tabelle darüber nachdenken, wie breit die einzelnen Spalten werden sollen.

Bei mechanischen Schreibmaschinen besteht das Setzen der Tabulatoren einfach aus dem Anbringen von Hindernissen an der Wagenlaufleiste. Durch das Drücken der Tabulatortaste wird die Wagensperre gelöst und der Wagen saust vorwärts bis zum nächsten Hindernis.

Die Tabulatormethode zum Tabellensatz wird auch von vielen Dokumentverarbeitungssystemen in mehr oder minder abgewandelter und verfeinerter Form zur Verfügung gestellt. Das ist insbesondere auch bei den Systemen  $\text{T}_{\text{E}}\text{X}$  und – auf deutlich verschiedene Weise –  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  der Fall.

### 5.1.1 Tabulatoren in $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

Eine Tabulator-Tabelle wird in  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  mit dem Kommando `\begin{tabbing}` begonnen und mit `\end{tabbing}` beendet. Die Tabulatorpositionen werden mit dem Befehl `\=` gesetzt. Der Befehl `\>` entspricht dem Druck auf die Tabulatortaste, der bewirkt, daß die nächste Tabulatorposition angesteuert wird. Im Inneren einer Tabulator-Tabelle ist der automatische Zeilenumbruch abgeschaltet; die Zeilen werden vielmehr durch den Befehl `\\` explizit beendet.

In den folgenden Beispielen befindet sich immer links die Eingabe für die rechts gezeigte Tabelle.

```

\begin{tabbing}
yellow \= jaune \= gelb \\
blue \> bleu \> blau \\
green \> vert \> gr"un
\end{tabbing}

```

yellow jaune gelb  
blue bleu blau  
green vert grün

Die Tatsache, daß in dem Beispiel die Tabellenbeschreibung selbst tabellenartig mit senkrecht ausgerichteten Kommandos gestaltet ist, ist beim Lesen der  $\LaTeX$ -Dokumentbeschreibung nützlich, aber nicht unbedingt erforderlich, um den gewünschten Effekt zu erzielen. Die Tabelle könnte auch völlig unordentlich beschrieben werden:

```

\begin{tabbing}
yellow \= jaune
\= gelb \\
blue
\> bleu \>
blau \\
green\> vert \> gr"un
\end{tabbing}

```

yellow jaune gelb  
blue bleu blau  
green vert grün

Glücklicherweise waren im obigen Beispiel die Wörter in der ersten Zeile, wo die Tabulatorpositionen gesetzt werden, länger als in den anderen beiden Zeilen. Was ist aber, wenn das nicht der Fall ist?

```

\begin{tabbing}
blue \= bleu \= blau \\
green \> vert \> gr"un \\
yellow \> jaune \> gelb
\end{tabbing}

```

blue bleu blau  
greenvert grün  
yellowjaune gelb

Man sieht, daß der  $n$ te Befehl `\>` immer eine Verschiebung zur  $n$ ten Tabulatorposition bewirkt, egal, ob diese noch rechts oder bereits links von der momentan erreichten Position liegt. Dies ist ein Unterschied zur Schreibmaschine, wo ein Druck auf die Tabulatortaste immer nach rechts zur nächsten noch nicht erreichten Position rücken läßt.

Um Tabellen setzen zu können, in denen nicht die erste Zeile die längsten Einträge enthält, kann man das `\kill`-Kommando benutzen. Eine Zeile, die mit `\kill` statt `\\` endet, erscheint nicht im formatierten Dokument, sondern dient nur dazu, Tabulatorpositionen zu setzen.

```

\begin{tabbing}
yellow \= jaune \= gelb \kill
blue \> bleu \> blau \\
green \> vert \> gr"un \\
yellow \> jaune \> gelb
\end{tabbing}

```

blue bleu blau  
green vert grün  
yellow jaune gelb

Es ist auch möglich, feste Maße für die einzelnen Spalten zu benutzen. Das Kommando `\hspace{len}` erzeugt einen leeren Raum der Länge  $len$ . Die Einheit  $em$  bezeichnet die Breite des Buchstaben „M“ im gerade aktiven Font.

```

\begin{tabbing}
\hspace{4em}\=\hspace{4em}\=\kill
blue \> bleu \> blau \\\
green \> vert \> gr"un \\\
yellow \> jaune \> gelb
\end{tabbing}

```

blue	bleu	blau
green	vert	grün
yellow	jaune	gelb

Der Befehl `\>` bewirkt, daß eine Tabulatorposition angesprungen wird. Darauf folgender Text beginnt normalerweise an der Tabulatorposition und erstreckt sich von dort aus nach rechts. Wenn er aber den Befehl `\'` enthält, dann wird er so weit nach links geschoben, daß `\'` an die Tabulatorposition zu liegen kommt, wie im folgenden Beispiel zu sehen ist.

```

\begin{tabbing}
Die Jahreszeiten: \= Fr"uhling, \\\
\> Sommer, \\\
\> Herbst, \\\
\> und \\' Winter
\end{tabbing}

```

Die Jahreszeiten: Frühling,
Sommer,
Herbst,
und Winter

Für die meisten Tabellen, die wir bisher gesehen haben, ist es empfehlenswerter, den in Abschnitt 5.2 beschriebenen Mechanismus zur logischen Tabellenbeschreibung zu benutzen, weil dort die Spaltenbreiten automatisch bestimmt werden. Der Tabulatormechanismus sollte dagegen eher in Fällen benutzt werden, wo die Spaltenstruktur unklar ist oder sich ständig verändert, wie im Falle des Setzens von Algorithmen oder Programmtexten. Es ist nämlich (innerhalb der `tabbing`-Umgebung) jederzeit möglich, Tabulatorpositionen umzudefinieren oder neue hinzuzufügen.

```

\begin{tabbing}
if \= $i > j$ \\\
\> then \= $i := j$; \\\
\> \> $j := i * i$ \\\
\> else \> $j := i$; \\\
\> \> $i := j * j$ \\\
fi; \\\
while \= $i < k$ do \\\
\> $a[i] := i$; \\\
\> $i := i + 1$ \\\
od
\end{tabbing}

```

```

if  $i > j$ 
then  $i := j$ ;
 $j := i * i$ 
else  $j := i$ ;
 $i := j * j$ 
fi;
while  $i < k$  do
 $a[i] := i$ ;
 $i := i + 1$ 
od

```

Bei solchen Programmtexten entstehen leicht tiefe Einrückungen, so daß Zeilen mit vielen `\>`-Kommandos in Folge beginnen müßten. Ein Kommando `\+` bewirkt, daß die folgenden Zeilen eine Tabulatorposition weiter rechts beginnen. Weitere `\+`-Kommandos rücken den Zeilenanfang immer weiter nach rechts. Der Effekt der `\+`-Kommandos wird durch `\-`-Kommandos dauerhaft aufgehoben. Ein `\<`-Kommando am Zeilenanfang hebt ein `\+`-Kommando lokal für diese Zeile auf.

```

\begin{tabbing}
if \= $i > j$           \+ \\\           if  $i > j$ 
    then \= $i := j$;   \+ \\\           then  $i := j$ ;
        $j := i * i$   \\\               $j := i * i$ 
    \< else \> $j := i$; \\\              else  $j := i$ ;
        $i := j * j$   \- \- \\\          $i := j * j$ 
fi;                      \\\              fi;
while \= $i < k$ do     \+ \\\           while  $i < k$  do
    $a[i] := i$;        \\\               $a[i] := i$ ;
    $i := i + 1$        \- \\\            $i := i + 1$ 
od
\end{tabbing}

```

Im Gegensatz zu den meisten anderen  $\LaTeX$ -Umgebungen dürfen `tabbing`-Umgebungen nicht ineinander geschachtelt werden. Der Effekt einer Schachtelung kann jedoch durch die Befehle `\pushtabs` und `\poptabs` simuliert werden, die die Werte der Tabulatorpositionen abspeichern und wieder hervorholen.

```

\begin{tabbing}
if \= $i > j$ \\\
    \> then \= $i := j$; \\\
    \>     \> $j := i * i$ \\\
\pushtabs
(* \= Dieses Programm \\\
    \> ist ziemlich sinnlos. *) \\\
\poptabs
    \> else \> $j := i$; \\\
    \>     \> $i := j * j$ \\\
fi;
\end{tabbing}

```

Die Befehle `\pushtabs` und `\poptabs` verhalten sich wie öffnende und schließende Klammern, d. h. in der Konstellation

```
... \pushtabs ... \pushtabs ... \poptabs ... \poptabs ...
```

bezieht sich das erste `\poptabs` auf das zweite `\pushtabs` und umgekehrt.

Die meisten der Tabulatorbefehle `\=`, `\>`, `\+`, `\-` usw. haben nur innerhalb der `tabbing`-Umgebung die oben beschriebene Bedeutung. Außerhalb sind sie undefiniert oder bedeuten etwas ganz anderes. Das Kommando `\-` z. B. zeigt normalerweise eine potentielle Trennstelle in einem Wort an.

Im Inneren der `tabbing`-Umgebung grenzt jeder der Tabulatorbefehle `\=`, `\>`, `\>` usw. eine Gruppe ab. Ein Schriftwechselkommando wirkt also nur bis zum nächsten derartigen Befehl. Daher ist es umständlich, eine ganze Zeile in einer besonderen Schriftart zu setzen; der Schriftwechselbefehl muß ständig wiederholt werden.

Eine ganze Spalte mit einem Befehl in einer besonderen Schrift zu setzen, ist gleichfalls unmöglich; es gibt ja noch nicht einmal eine Stelle in der Dokumentbeschreibung, wo man einen solchen Befehl hinschreiben könnte.

Außer diesen Schwächen gibt es noch weitere Defizite. So ist es nicht möglich, Text um eine Tabulatorposition herum zu zentrieren. Die in einer `tabbing`-Umgebung gesetzten Tabulatorpositionen gehen an ihrem Ende verloren; es gibt in der nächsten `tabbing`-Umgebung keine Erinnerung mehr daran.

## 5.1.2 Tabulatoren im *FrameMaker*

Im *FrameMaker* gibt es keine besondere Umgebung für die Benutzung von Tabulatoren wie z. B. die `tabbing`-Umgebung in  $\text{\LaTeX}$ . Vielmehr können Tabulatoren überall gesetzt und angesprungen werden.

### Arten von Tabulatoren

Im *FrameMaker* gibt es vier verschiedene Arten von Tabulatoren: „left“, „right“, „center“ und „decimal“. Die Art „left“ ist die klassische Schreibmaschinenart, wie sie auch in  $\text{\LaTeX}$  verwirklicht ist. Text, der an einen „left“-Tabulator gesetzt wird, beginnt an der Tabulatorposition und erstreckt sich von dort aus nach rechts. An einen „right“-Tabulator wird Text so gesetzt, daß er genau an der Tabularposition endet, und an einen „center“-Tabulator so, daß er um die Position herum zentriert ist.

An einen (standardmäßigen) „decimal“-Tabulator sollte nur Text gesetzt werden, der genau einen Punkt enthält. Der Text wird dann so positioniert, daß der Punkt genau an der Tabulatorposition liegt. Damit können Zahlen mit Dezimalpunkt wie z. B. 123.4 und 3.14159 so ausgerichtet werden, daß die Punkte genau übereinanderliegen. Einzelne oder alle Dezimaltabulatoren können umdefiniert werden, so daß sie auf etwas anderes als einen Punkt achten, z. B. auf ein Komma oder ein beliebiges anderes Zeichen.

Im Vergleich zu diesen Möglichkeiten ist  $\text{\LaTeX}$  weniger mächtig. Dort gibt es nur „left“-Tabulatoren. Mit dem Befehl `\'` können „right“- und „decimal“-Tabulatoren immerhin simuliert werden. Dagegen gibt es nichts, was einem „center“-Tabulator entsprechen würde.

### Das Setzen von Tabulatoren

Das Setzen von Tabulatoren, das in  $\text{\LaTeX}$  mit dem Kommando `\=` durchgeführt wird, erfolgt im *FrameMaker* auf visuellem Wege. Quer über der Seite kann ein Lineal dargestellt werden. Über dem Lineal befinden sich vier Felder mit Symbolen für die vier Tabulatorarten. Ein neuer Tabulator wird an eine bestimmte Position gesetzt, indem eine Kopie des Tabulatorsymbols aus dem entsprechenden Feld mit

der Maus an die gewünschte Stelle am Lineal gezogen wird. Ein Tabulator wird durch Bewegen mit der Maus verschoben und durch Wegziehen vom Lineal nach unten gelöscht. Bei der Verschiebung eines Tabulators wird bereits existierender Text, der nach diesem Tabulator ausgerichtet ist, mitverschoben.

Außer diesem visuellen Verfahren gibt es noch die Möglichkeit, mit Dialogboxen zu arbeiten. Damit kann man die Art eines existierenden Tabulators umdefinieren, das Ausrichtungszeichen eines „decimal“-Tabulators ändern, einen neuen Tabulator an eine durch einen Zahlenwert bestimmte Position legen oder mehrere Tabulatoren auf einmal mit regelmäßigen Zwischenräumen setzen.

## Das Anspringen von Tabulatorpositionen

Während der Texteingabe wird durch Druck auf die „Tab“-Taste die nächste Tabulatorposition angesprungen. Das entsprechende  $\text{\LaTeX}$ -Kommando war  $\backslash>$ . Unabhängig von der Lage des Einfügepunktes wird beim *iten* „Tab“ in einer Zeile zum *iten* Tabulator gesprungen. Dies entspricht dem Verhalten von  $\text{\LaTeX}$ , weicht aber von dem Verhalten der Schreibmaschine ab, wo es nur Sprünge nach rechts gibt.

In der bisherigen Beschreibung haben wir angenommen, daß zuerst Tabulatoren gesetzt werden und dann erst Text mit „Tab“-Zeichen eingegeben wird. In einen Text können aber auch „Tab“-Zeichen eingefügt werden, wenn noch gar keine oder zu wenige Tabulatoren gesetzt sind. In diesem Falle haben die „Tab“-Zeichen (zunächst) keinen Effekt. Ihre Lage im Text kann jetzt schon am Bildschirm dargestellt werden. Wenn später dann Tabulatoren für diesen Text gesetzt werden, dann werden die „Tab“-Zeichen aktiv und bewirken eine entsprechende Ausrichtung von Textteilen. Dies ist ein Unterschied zu  $\text{\LaTeX}$ , wo es ein Fehler ist,  $\backslash>$  zu benutzen, ohne daß vorher mit  $\backslash=$  eine Tabulatorposition festgelegt wurde.

## Tabulatoren und Paragraphenarten

Bisher haben wir das Tabulieren im *FrameMaker* so beschrieben, als ob es immer spontan für einen bestimmten Text erfolgen würde. Das ist tatsächlich möglich, aber es geht auch anders.

Zu den Eigenschaften einer *FrameMaker*-Paragraphenart gehören auch Tabulatorsetzungen. Diese Tabulatorsetzungen gelten dann für alle Paragraphen dieser Art. Damit ist es möglich, von einer Tabulatorsetzung zu einer anderen zu wechseln und wieder zurück wie mit den  $\text{\LaTeX}$ -Befehlen  $\backslash\text{pushtabs}$  und  $\backslash\text{poptabs}$ . Außerdem können Tabulatorsetzungen über große Entfernungen im Dokument hinweg im Gedächtnis behalten und wiederhergestellt werden, was in  $\text{\LaTeX}$  nicht ohne weiteres möglich ist.

## 5.2 Logisch beschriebene Tabellen

Neben der Tabulatormethode bieten viele Dokumentverarbeitungssysteme eine mehr logisch orientierte Methode zur Tabellenbeschreibung an. Eine logische Tabelle besteht aus vielen rechteckigen Kästchen, den *Tabelleneinträgen*, die zweidimensional in *Zeilen* und *Spalten* angeordnet sind.

Die Tabellenbeschreibung erfolgt meist zeilenweise; die Autoren geben die einzelnen Zeilen der Tabelle ein und bezeichnen dabei jeweils den Übergang zu einer neuen Spalte. Dazu gehört eine *Musterzeile*, die angibt, wie alle Zeilen der Tabelle nach der Formatierung aussehen sollen.

Manche Systeme bieten als Alternative die spaltenweise Beschreibung von Tabellen an. Die Autoren geben in diesem Fall die einzelnen Spalten der Tabelle ein und bezeichnen im Innern der Spalte jeweils die Übergänge zu einer neuen Zeile. Dazu gehört dann eine *Musterspalte*.

In einer zeilenweisen Beschreibung ist das Vertauschen zweier Zeilen sehr einfach, dagegen das Vertauschen zweier Spalten recht aufwendig. In einer spaltenweisen Beschreibung ist es umgekehrt.

Unabhängig von der Art der Beschreibung berechnet das System für jede Spalte das Maximum der Breiten der einzelnen Einträge in dieser Spalte; dies wird die Breite der gesamten Spalte. Ähnlich wird die Höhe der einzelnen Zeilen ermittelt.

Das System  $\text{T}_{\text{E}}\text{X}$  bietet sowohl die zeilenweise Beschreibung mit dem Kommando `\halign` an, als auch die spaltenweise Beschreibung mit `\valign` (siehe [Knu86b] und auch [BK89]). In  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  gibt es nur die zeilenweise Beschreibung mit einem Mechanismus, der sich deutlich von `\halign` unterscheidet. Im folgenden wird dieser  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ -Mechanismus beschrieben.

### 5.2.1 Logisch beschriebene Tabellen in $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

Eine  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ -Tabellenbeschreibung beginnt mit `\begin{tabular}` gefolgt von einer Musterzeile. Dann folgen die einzelnen Zeilen der Tabelle. Innerhalb der Zeilen wird der Übergang zu einer neuen Spalte durch das Zeichen `&` angezeigt. Die Zeilen werden wie im Tabulator-Mechanismus durch `\\` beendet. Das Ende der ganzen Tabelle wird durch `\end{tabular}` angezeigt.

Als erstes betrachten wir wieder die Farbtabelle wie am Anfang von Abschnitt 5.1. Im Gegensatz zu dort werden die Spaltenbreiten automatisch berechnet, so daß zwischen je zwei Spalten ein bestimmter Abstand bleibt. Die Musterzeile `1 1 1` bedeutet zunächst nur, daß die Tabelle drei Spalten hat.

```
\begin{tabular}{1 1 1}
blue   & bleu  & blau  \\
green  & vert  & gr"un \\
yellow & jaune & gelb  \\
\end{tabular}
```

blue	bleu	blau
green	vert	grün
yellow	jaune	gelb

In der Musterzeile werden die einzelnen Spalten durch die Buchstaben `l`, `c` oder `r` bezeichnet. Die Einträge einer mit `l` bezeichneten Spalte werden linksbündig gesetzt, die einer mit `c` bezeichneten zentriert und die einer `r`-Spalte rechtsbündig. Diese Flexibilität wäre im Tabulatormechanismus nur durch Tricks zu erreichen.

```
\begin{tabular}{l c r}
blue   & bleu  & blau  \\
green  & vert  & grün  \\
yellow & jaune & gelb  \\
\end{tabular}
```

blue	bleu	blau
green	vert	grün
yellow	jaune	gelb

Tabellen sind oft durch horizontale und vertikale Linien gegliedert. Das ist kein Problem; im folgenden Beispiel werden die senkrechten Linien in der Musterzeile beschrieben, während die horizontale Linie durch `\hline` erzeugt wird.

```
\begin{tabular}{l | l | l}
english & fran\c{c}ais & deutsch \\
\hline
blue    & bleu        & blau   \\
green   & vert        & grün   \\
yellow  & jaune       & gelb   \\
\end{tabular}
```

english	français	deutsch
blue	bleu	blau
green	vert	grün
yellow	jaune	gelb

Es ist auch möglich, in der Musterzeile Textstücke anzugeben, die in allen Zeilen gleichermaßen vorkommen sollen. Diese Textstücke werden durch `@` eingeleitet.

```
\begin{tabular}{ @{\it englisch } 1
                  @{\it franz"osisch } 1
                  @{\it deutsch } 1 }
blue   & bleu  & blau  \\
green  & vert  & grün  \\
yellow & jaune & gelb  \\
\end{tabular}
```

*englisch* blue = *französisch* bleu = *deutsch* blau  
*englisch* green = *französisch* vert = *deutsch* grün  
*englisch* yellow = *französisch* jaune = *deutsch* gelb

Das folgende Abschlußbeispiel zeigt einige andere Möglichkeiten. Eine Tabelle kann Spalten fester Breite *len* enthalten, die durch `p{len}` in der Musterzeile bezeichnet werden. Die Einträge in einer solchen Spalte werden automatisch in Zeilen eingeteilt.

Während die von dem Befehl `\hline` erzeugten waagerechten Linien über die ganze Breite der Tabelle hinweggehen, bewirkt der Befehl `\cline{i-j}` eine Linie, die nur die Spalten *i* bis *j* überstreicht.

Eine Tabelle kann in manchen Zeilen Einträge beinhalten, die über mehrere Spalten hinwegreichen. Diese werden durch das Kommando `\multicolumn` mit drei Argumenten spezifiziert. Das erste Argument gibt an, wie viele Spalten überdeckt werden. Das zweite ist `l`, `c` oder `r` und gibt an, wie der Eintrag ausgerichtet werden soll. Das dritte Argument ist der Eintrag selbst. Ein `\multicolumn`, das nur eine Spalte überspannt, kann benutzt werden, um die Standardausrichtung für diese Spalte zu umgehen.

Beachten Sie, wie die senkrechten Linien in die Beschreibung der `\multicolumn`-Einträge mit aufgenommen werden müssen. Beachten Sie auch, daß in dem Beispiel der Preis aus zwei Spalten besteht, eine für die Vorkomma- und eine für die Nachkommastellen. Das zweite `\multicolumn` in der ersten Tabellenzeile dient dazu, das Komma zwischen diesen beiden Spalten zu unterdrücken.

```
\begin{tabular}{| l | l | r @{,} l || p{4cm} | } \hline
\multicolumn{2}{|c|}{Ware} & \multicolumn{2}{c|}{} & \\
\multicolumn{1}{|c|}{Art} & \multicolumn{1}{c|}{Menge} & & & \\
\multicolumn{2}{c|}{Preis (DM)} & & & \\
\multicolumn{1}{c|}{Bemerkungen} & & & & \\ \hline \hline
Nagel & St"uck & 0 & 001 & Einzeln billig. \\
& 100\,g & 0 & 79 & Daher meist in gr"o"seren \\
& & & & Mengen verkauft. \\
Hammer & St"uck & 2 & 98 & Ideal zum H"ammern. \\
Leim & Topf & 11 & 90 & Festkleben ist besser \\
& & & & als festnageln! \\ \hline
\end{tabular}
```

Ware		Preis (DM)	Bemerkungen
Art	Menge		
Nagel	Stück	0,001	Einzeln billig.
	100 g	0,79	Daher meist in größeren Mengen verkauft.
Hammer	Stück	2,98	Ideal zum Hämmern.
Leim	Topf	11,90	Festkleben ist besser als festnageln!

Zusammenfassend ist zu sagen, daß das `tabular`-Konzept von  $\text{\LaTeX}$  recht mächtig ist, aber nicht ganz einfach zu benutzen. Außerdem sind gewisse wünschenswerte Funktionalitäten nicht vorgesehen. Man kann z. B. Tabelleneinträge zwar horizontal, aber nicht vertikal zentrieren; es ist also (ohne irgendwelche Tricks) unmöglich, im obigen Beispiel „Leim“ oder „Topf“ in ihrem Kasten tiefer zu setzen.

Es gibt auch keine Möglichkeit, ein Kommando auf eine ganze Zeile oder eine ganze Spalte einwirken zu lassen, also z. B. eine ganze Zeile oder Spalte mit einem

einzigem Kommando kursiv zu setzen. Ein Schriftwechselkommando am Anfang einer Zeile wirkt nämlich nur in der ersten Spalte, da jedes Tabellenfeld eine eigene Gruppe bildet. Kommandos, die eine ganze Spalte betreffen, könnten in der Musterzeile gegeben werden, aber das ist nicht erlaubt.

## 5.2.2 Logisch beschriebene Tabellen in *FrameMaker*

Logisch beschriebene Tabellen, im folgenden kurz Tabellen genannt, sind in *FrameMaker* Gleitobjekte, die im Text eingebettet oder unabhängig vom Text in der Nähe eines Ankers erscheinen. Sie können eine Überschrift besitzen, die von der eigentlichen Tabelle unabhängig ist. Im Gegensatz zu  $\text{\LaTeX}$ , wo alle Tabellenzeilen im Prinzip gleichwertig sind, besteht eine *FrameMaker*-Tabelle aus *Kopfzeilen*, *Rumpfzeilen* und *Fußzeilen*. Die Kopfzeilen beinhalten Überschriften für die Spalten, während die Rumpfzeilen den eigentlichen Inhalt tragen. Die meisten Tabellen haben gar keine Fußzeilen. In langen Tabellen können die Spaltenüberschriften in einer Fußzeile wiederholt werden.

Von der Handhabung her entspricht jedes Tabellenfeld einer Textspalte. Es hat also eine feste Breite, während sich die Höhe am Inhalt orientiert. Text in einem Tabellenfeld wird automatisch umgebrochen, was den mit `p` bezeichneten Spalten fester Breite von  $\text{\LaTeX}$  entspricht. Tabellenfelder können alles enthalten, was auch eine Spalte auf einer Seite enthalten kann, also Folgen von Paragraphen und insbesondere wieder Tabellen.

### Tabellenformate

Die graphische Erscheinungsform einer *FrameMaker*-Tabelle richtet sich nach einem Tabellenformat. Dieses Prinzip haben wir bereits bei den Seiten kennengelernt, die durch Hintergrundseiten beschrieben werden, bei Paragraphen und Paragraphenarten sowie bei Verweisen und Verweisformaten.

Jeder konkreten Tabelle ist ein Tabellenformat zugeordnet, das das Aussehen der Tabelle generisch beschreibt. Es gibt einen Katalog von Tabellenformaten, der von den Autoren durch die Definition neuer Formate oder das Umdefinieren existierender Formate verändert werden kann. Das Aussehen einer einzelnen Tabelle kann unabhängig vom gewählten Format nachträglich geändert werden.

Durch ein Tabellenformat können nur generische Eigenschaften von Tabellen beschrieben werden, also Eigenschaften, die ganzen Zeilen oder Spalten zukommen. Es kann nicht spezifiziert werden, daß ein bestimmtes Feld besonders hervorgehoben werden soll, z. B. durch einen fetten Rand. Solche Veränderungen können nur nachträglich an einer konkret existierenden Tabelle vorgenommen werden.

Ein Tabellenformat beschreibt vier verschiedene Gruppen von Eigenschaften, die wir in den folgenden Abschnitten kurz beschreiben werden: „Default“-Eigenschaften, Linien („Ruling“), Schattierung („Shading“) und Positionierung (im *Frame-*

*Maker* „Basic“ genannt). Wenn das Tabellenformat einer bereits existierenden Tabelle geändert wird, dann behält die Tabelle ihre Default-Eigenschaften, während sich die Linien, die Schattierung und die Positionierung nach dem neuen Format richten.

## Default-Eigenschaften einer Tabelle

Zu den Default-Eigenschaften gehören die Anzahl der Kopf-, Rumpf- und Fußzeilen, die Zahl und Breite der Spalten, das Format des ersten Paragraphen im Titel (wenn es einen Titel gibt) sowie das Format des ersten Paragraphen in den Kopf-, Rumpf- und Fußfeldern jeder Spalte.

## Linien

Das Tabellenformat beschreibt auch die Trennlinien zwischen den einzelnen Zeilen und Spalten der Tabelle sowie die Linien am Rand der Tabelle. Nach der üblichen Philosophie des *FrameMaker* hat jede Linie ein *Linienformat*. Es gibt einige vordefinierte Formate, z. B. „Thin“ (dünn), „Medium“ (mittel), „Thick“ (dick), „Double“ (Doppellinie) sowie „None“ (nicht sichtbar). Die Autoren können existierende Formate umdefinieren oder neue hinzudefinieren. Zu einem Linienformat gehören unter anderem Angaben über die Anzahl und Dicke der Linien sowie ihren Abstand, wenn es mehrere sind.

Es können die folgenden Arten von Linien definiert werden:

- Spaltentrennlinien, das sind die Trennlinien zwischen zwei Spalten (nicht die ganz am Rand der Tabelle). Die Spaltentrennlinien können in zwei Gruppen eingeteilt werden, denen getrennt Linienformate zugewiesen werden können. Die erste Gruppe kann z. B. nur aus der ersten Trennlinie (von links gezählt) bestehen, oder aus allen mit gerader Nummer. Die zweite Gruppe besteht immer aus allen anderen Trennlinien.
- Rumpfzeilentrennlinien, das sind die Trennlinien zwischen zwei Rumpfzeilen. Sie können ähnlich wie die Spaltentrennlinien in zwei Gruppen eingeteilt werden.
- die Trennlinie zwischen Kopf und Rumpf und die zwischen Rumpf und Fuß, wenn ein solcher existiert. Diese zwei Trennlinien erhalten dasselbe Format.
- Zeilentrennlinien im Innern des Kopfes und im Innern des Fußes. Allen diesen Trennlinien kann nur ein Format zugewiesen werden.
- Die vier Außenlinien am oberen, unteren, linken und rechten Rand der Tabelle können unabhängig voneinander festgelegt werden.

Die im Tabellenformat enthaltenen Liniendefinitionen können später für konkrete Tabellen abgeändert werden. Damit ist es z. B. möglich, einzelne Tabellenfelder fett zu umranden.

## Schattierung

Für Schattierungen gibt es keine Schattierungsformate, sondern jedem zu schattierenden Gebiet wird direkt eine Prozentzahl zugeordnet, von 0% für weiß bis 100% für schwarz. Die Definition der Schattierung erfolgt entweder spaltenweise oder zeilenweise. Bei spaltenweiser Definition können die Spalten in zwei Gruppen eingeteilt werden. Bei zeilenweiser Definition erhalten die Kopf- und Fußzeilen eine gemeinsame Schattierung, während die Rumpfzeilen davon unabhängig in zwei Gruppen eingeteilt werden können.

## Positionierung

Die Positionierung einer Tabelle legt fest, ob sie als ganzes linksbündig, rechtsbündig oder zentriert ausgegeben wird, wie viel Platz vor und nach der Tabelle gelassen wird und ob und wie sie durch Seitenwechsel geteilt werden darf. Außerdem kann definiert werden, welche Abstände der Inhalt eines Tabellenfeldes von den vier Rändern des Feldes einhalten soll.

Zu dieser Gruppe von Eigenschaften gehört noch die Angabe, ob automatisch berechnete Numerierungen in den Tabellenfeldern auf einem zeilenweisen oder spaltenweisen Durchlauf der Tabelle beruhen sollen.

## Erstellen einer Tabelle

Zum Erstellen einer Tabelle gibt es verschiedene Methoden, die wir im folgenden beschreiben werden.

Eine neue Tabelle kann unter Angabe eines Formats erzeugt werden. Das Format regelt die Erscheinungsform der neuen Tabelle, die mangels Inhalt zunächst aus leeren Feldern besteht. Wie oben beschrieben, gehören zum Format unter anderem Defaultwerte für die Anzahl der Spalten und Zeilen (getrennt nach Kopf-, Rumpf- und Fußzeilen), die noch während des Entstehungsprozesses der Tabelle auf andere Werte gesetzt werden können.

Eine andere Methode ist, eine existierende Tabelle zu kopieren – entweder mit Inhalt oder ohne. Im letzteren Fall wird nur die graphische Struktur der Tabelle kopiert; die Felder bleiben leer.

Eine leere Tabelle kann gefüllt werden, indem die Felder nacheinander selektiert und mit Inhalt belegt werden. Das Selektieren kann mit der Maus geschehen. Es gibt aber auch Tastaturkommandos, die zum nächsten Feld nach rechts, links, oben oder unten weiterschalten. Das Kommando zum Weiterschalten nach rechts ist besonders einfach, nämlich ein Druck auf die „Tab“-Taste.

Mit dem Kommando „Control-Return“ entsteht eine neue leere Zeile unter derjenigen, in der gerade gearbeitet wurde. Unter Benutzung von „Tab“ und „Control-Return“ kann also eine Tabelle bequem zeilenweise eingegeben werden.

## Umwandeln von Text in eine Tabelle und umgekehrt

Eine andere Methode zur Tabellenerstellung ist die Umwandlung von existierendem Text, also einer Folge von Paragraphen, in eine Tabelle. Für eine solche Umwandlung gibt es verschiedene Möglichkeiten: Jeder Paragraph kann ein eigenes Tabellenfeld werden, wobei die Anzahl der Spalten vorher festgelegt wird. Oder jeder Paragraph wird zu einer Tabellenzeile, wobei die Einteilung in Felder auf Wunsch durch „Tab“-Zeichen, eine gewisse Mindestzahl von Leerzeichen oder ein bestimmtes Zeichen wie z. B. ein Komma erfolgt. Der Umwandlungsprozeß, bei dem Paragraphen zu Zeilen werden und Felder durch „Tab“-Zeichen beendet werden, ist besonders interessant, da er aus einer Tabulator-Tabelle eine echte Tabelle ähnlicher Gestalt erzeugt.

Bei der Umwandlung einer Tabelle in gewöhnlichen Text wird jedes Tabellenfeld zu einem Paragraphen. Die Felder werden auf Wunsch zeilenweise oder spaltenweise durchlaufen. Die Umwandlungsprozesse von Text nach Tabelle und umgekehrt sind also nicht invers zueinander. Es ist z. B. nicht möglich, aus einer echten Tabelle eine entsprechende Tabulator-Tabelle zu erzeugen.

## Bearbeiten von Tabellen

Die Inhalte einzelner Felder können wie gewöhnlicher *FrameMaker*-Text bearbeitet werden. Dazu kommt die Möglichkeit, ganze Felder, ganze Zeilen oder Spalten oder die ganze Tabelle zu selektieren. Dieses Selektieren ist die erste Stufe für Lösch-, Verschiebe- oder Kopieraktionen. Bei diesen Aktionen wird jeweils gefragt, ob sie sich nur auf die Inhalte des selektierten Teils beziehen oder auf den ganzen Teil mit Inhalt. Das macht einen Unterschied; man kann z. B. entweder den Inhalt einer Spalte in eine andere Spalte kopieren, wobei die Spaltenzahl gleichbleibt, oder die Spalte selbst mitsamt Inhalt kopieren, wobei die Spaltenzahl um eins zunimmt.

Das Kopieren oder Verschieben von Zeilen und Spalten geht nicht nur innerhalb einer Tabelle, sondern auch zwischen verschiedenen Tabellen, allerdings nur, wenn sie in der Struktur zueinander passen. Eine Zeile z. B. kann aus einer Tabelle in eine andere nur verschoben werden, wenn beide Tabellen dieselbe Spaltenzahl haben.

## Regelung der Spaltenbreite und Zeilenhöhe

Durch das Tabellenformat wird jeder Spalte eine feste Breite zugewiesen. Der Inhalt der Tabellenfelder unterliegt dem automatischen Zeilenumbruch, so daß die festgelegte Spaltenbreite nicht überschritten wird. Im Gegensatz dazu macht das Format keine Aussage über die Höhe der Zeilen. Die Höhe, die ein einzelnes Feld benötigt, richtet sich nach der Menge seines Inhalts. Die Höhe einer Zeile wird automatisch so bestimmt, daß alle Felder der Zeile eine ausreichende Höhe haben. Dieses Verhalten entspricht dem der mit  $p$  bezeichneten Spalten fester Breite in  $\text{\LaTeX}$ .

Die Spaltenbreiten einer existierenden Tabelle können auf mehrere Weisen verändert werden. Am einfachsten ist eine visuell gesteuerte Anpassung durch Verschie-

ben von Spaltengrenzen mit der Maus. Dabei kann gewählt werden, ob eine Verbreiterung einer Spalte auf Kosten ihrer Nachbarspalte geht oder eine Verbreiterung der ganzen Tabelle bewirkt. Eine Spaltenbreite kann außerdem von einer anderen Spalte, die auch in einer anderen Tabelle liegen kann, kopiert werden.

Außer diesen visuell orientierten Verfahren gibt es ein Dialogfenster zur Breitendefinition, das sechs verschiedene Operationen erlaubt.

- Die Breiten der selektierten Spalten werden durch eine explizite Längenangabe definiert.
- Die Breiten der selektierten Spalten werden mit einem bestimmten Faktor multipliziert.
- Die Breiten aller selektierten Spalten werden auf die Breite einer bestimmten Spalte gesetzt.
- Die selektierten Spalten werden gleich breit. Die Summe der Breiten hat einen vorgegebenen Wert.
- Die Breiten der selektierten Spalten werden mit einem Faktor multipliziert, der von *FrameMaker* so berechnet wird, daß die Summe der Breiten einen vorgegebenen Wert hat.
- Die selektierten Spalten werden gerade so breit, daß aller Text in diesen Spalten ohne Zeilenumbruch gesetzt werden kann. Dies entspricht (bis auf die Zeilenausrichtung) dem Verhalten von l-, c- oder r-Spalten in  $\LaTeX$ .

Im Gegensatz zu den Spaltenbreiten können die Zeilenhöhen nur wenig beeinflußt werden. Eine visuelle Beeinflussung durch Mausoperationen ist unmöglich. Eine Mindesthöhe kann als explizite Längenangabe vorgeschrieben werden. Andere Operationen zur Festlegung der Zeilenhöhe sind nicht vorgesehen.

## Verschmelzen und Zerlegen von Feldern

Es ist möglich, mehrere benachbarte Felder zu einem zu verschmelzen. Der Inhalt des Gesamtfeldes ist eine Folge von Paragraphen, wobei jeder Paragraph einem Teilfeld entspricht. Diese Folge wird räumlich so positioniert, daß sie das gesamte Gebiet des verschmolzenen Feldes ausnutzt.

Die Verschmelzung mehrerer Felder in einer Zeile entspricht dem `\multicolumn`-Befehl von  $\LaTeX$ . Im Gegensatz zu  $\LaTeX$  ist man jedoch nicht auf Zeilen beschränkt; jede rechteckige Gruppe von Feldern kann zu einem verschmolzen werden.

Ein verschmolzenes Feld kann wieder in seine Teilfelder zerlegt werden. Dabei wandert der Inhalt jedoch nicht in die Felder zurück, aus denen er gekommen ist, sondern wird in das linke obere Teilfeld gefüllt, während die anderen Teilfelder leer bleiben. Die Hersteller des *FrameMaker* haben also nicht die Analogie zwischen dem Zerlegen eines Feldes in Teilfelder und der Umwandlung von Text in eine Tabelle erkannt, sondern die beiden Probleme unabhängig voneinander gelöst.