

# 1 Einführung

## 1.1 Dokumente und ihre Struktur

### 1.1.1 Dokumente und ihre Darstellung

Unter *Dokumenten* im allgemeinsten Sinne verstehen wir Bündel von *Informationen*, die in der Absicht erzeugt worden sind, daß sie von Menschen wahrgenommen werden. Beispiele dafür sind Reden, Filme, Bilder und Schriftdokumente. In diesem Buch werden wir uns allerdings auf die Betrachtung von Schriftdokumenten beschränken. Sie sind dadurch gekennzeichnet, daß sie nur optisch wahrgenommen werden (im Gegensatz zu Reden und Tonfilmen), sich während der Betrachtung nicht wesentlich verändern (im Gegensatz zu Filmen) und überwiegend aus Text bestehen (im Gegensatz zu Bildern). Schriftdokumente enthalten Bilder allenfalls als Zusatzerläuterungen zum Text (so ähnlich wie dieses Buch).

Während sich die bisher genannten Einschränkungen auf die äußere Erscheinungsform beziehen, machen wir noch eine weitere Einschränkung, die sich auf den Entstehungsprozeß des Dokuments bezieht. Wir werden nämlich (außer wenn es ausdrücklich anders gesagt wird) nur solche Schriftdokumente betrachten, die computergestützt erstellt und bearbeitet worden sind. Da wir im Rest des Buches nur von *computergestützten Schriftdokumenten* sprechen werden, werden wir sie kurz als „Dokumente“ bezeichnen. Das vorliegende Buch ist ein Dokument in diesem Sinn.

Da, wie eingangs erwähnt, ein Dokument ein Bündel von Informationen ist, sollten wir also zwischen Dokumenten an sich und ihren konkreten *Darstellungen* unterscheiden. An konkreten Darstellungen gibt es zunächst eine oder verschiedene *rechnerinterne* Darstellungen in Gestalt irgendwelcher Datenstrukturen im Speicher des Computers. Diese sind normalerweise nicht unmittelbar verständlich für Menschen und müssen erst in optisch wahrnehmbare *externe* Darstellungen übersetzt werden.

Eine externe Darstellung eines (Schrift-)Dokuments besteht aus optisch wahrnehmbaren Markierungen, die auf geeigneten Flächen entweder dauerhaft angebracht oder zeitweise sichtbar sind. Dauerhafte Darstellungen befinden sich heutzutage meist auf Papier, sind aber auch auf Stein (z. B. Grabsteinen), Metall oder Plastik (z. B. Hinweisschildern) zu finden. Zeitweise sichtbare Darstellungen sind auf Bildschirmen und Projektionsflächen möglich.

Ein computererzeugtes Dokument wie das vorliegende Buch existiert in vielen verschiedenen Darstellungen. Da sind zunächst die rechnerinternen, von denen es

mehrere verschiedene gibt, wie wir sehen werden. Eine davon ist die von den Autoren erstellte Dokumentbeschreibung. Davon kann es aus Sicherheitsgründen mehrere Kopien geben. Aus der Dokumentbeschreibung werden durch automatische Bearbeitungs- und Übersetzungsvorgänge andere, wesentlich verschiedene interne Darstellungen erzeugt. Diese können durch Ausgaberroutinen in externe Darstellungen auf dem Bildschirm oder auf Papier verwandelt werden. Eine bestimmte interne Darstellung ist die Grundlage des Druck- und Vervielfältigungsprozesses. Daraus entstehen viele verschiedene externe Darstellungen, nämlich die einzelnen Exemplare des Dokuments, die im Idealfall alle gleich aussehen. Durch den Einsatz von Fotokopierern können weitere externe Darstellungen entstehen, die gegenüber dem Original vergrößert oder verkleinert sein können.

Wegen der vielen verschiedenen Darstellungen ist die Frage nach der Identität von Dokumenten problematisch. Eine unveränderte Neuauflage eines Buches bietet sicherlich eine Darstellung desselben Dokuments wie die Erstauflage. Was ist aber mit einer in anderem Format gesetzten Neuauflage oder gar mit einer Neuauflage mit Fehlerkorrekturen? Was ist mit Büchern, die von ihren Lesern mit Randmarkierungen versehen worden sind?

In diesem Buch werden wir, um unnötiges Kopfzerbrechen zu vermeiden, auf diese Fragen nicht näher eingehen. Obwohl wir oft auch statt von Dokumentdarstellungen einfach von Dokumenten sprechen, sollten wir aber im Gedächtnis behalten, daß ein Dokument eine von seiner konkreten Darstellung unabhängige, sozusagen ideelle Existenz führt.

## 1.1.2 Physikalische und visuelle Erscheinungsform von Dokumenten

Wie bereits erwähnt, besteht eine externe Darstellung eines Dokuments aus Markierungen auf einem „Substrat“, d. h. einer geeigneten Fläche (meist Papier oder Bildschirm). Die Markierungen auf dem Substrat haben meßbare Eigenschaften, die zusammen die *physikalische Erscheinungsform* der externen Dokumentdarstellung ausmachen.

Insofern das Dokument überwiegend aus Text besteht, sind die „Markierungen“ im allgemeinen *Schriftzeichen*. Meßbare Eigenschaften einzelner Schriftzeichen sind ihre *graphischen Attribute* wie Breite, Höhe, Strichstärke, Neigung usw. Diese bestimmen die *typographischen Attribute* wie Schriftschnitt oder Laufweite. In Abschnitt 2.3 wird genauer auf die Eigenschaften von Schriftzeichen eingegangen.

Meßbare Eigenschaften von Gruppen von Schriftzeichen sind die Abstände der Zeichen in einem Wort und zwischen zwei Worten sowie zwischen zwei Zeilen. Meßbare Eigenschaften der Darstellung als ganzes sind z. B. die Länge der Zeilen und die Höhe des bedruckten Raumes auf einer Seite. Dazu kommen noch weniger leicht erfaßbare statistische Eigenschaften wie die durchschnittliche Schwärzung in einer Zeile, die von der Form der Schriftzeichen und ihren Abständen zueinander abhängt.

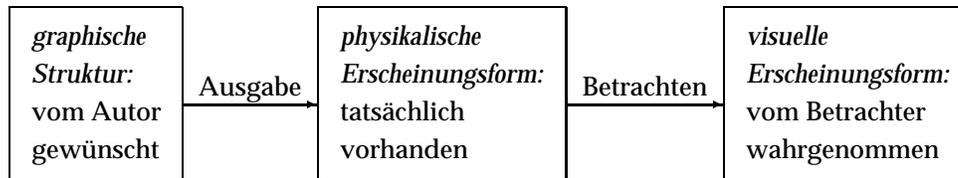


Abbildung 1.1: Dokumentstrukturen und -erscheinungsformen

Das menschliche visuelle System nimmt die graphischen Attribute der Schriftzeichen als *visuelle Attribute* wahr, den Durchmesser etwa als *wahrgenommene Größe*, die Strichstärke als *Gewicht* und die Neigung als *Schrägstellung*. Ebenso werden die objektiv meßbaren Beziehungen zwischen den Zeichen vom menschlichen Sehsystem als subjektive *visuelle Beziehungen* interpretiert. Die visuellen Attribute der Markierungen in einem extern dargestellten Dokument und ihre visuellen Beziehungen zueinander stellen die *visuelle Erscheinungsform* des Dokuments dar.

Dabei kann es durchaus Unterschiede zwischen der objektiven physikalischen und der subjektiv wahrgenommenen visuellen Erscheinungsform geben (optische Täuschungen).

### 1.1.3 Die graphische Struktur von Dokumenten

Wenn ein Dokument auf mehreren verschiedenen Druckern oder mehrmals hintereinander auf demselben Drucker ausgegeben wird, dann stimmen die dadurch entstehenden physikalischen Erscheinungsformen im allgemeinen nicht exakt überein. Die genaue Erscheinungsform hängt von potentiellen Störungen oder Fehlfunktionen des Druckers ab. Ein ungleichmäßig schneller Papierdurchzug zum Beispiel kann Zeilenhöhen und -abstände beeinflussen. Selbst bei optimal arbeitenden Druckern kann es Unterschiede geben, da Drucker verschiedener Hersteller sich manchmal in der Fähigkeit unterscheiden, kleine Details oder ungewöhnliche Zeichen darstellen zu können.

Um von diesen Unwägbarkeiten abstrahieren zu können, sprechen wir von einer idealisierten *graphischen Dokumentstruktur*, die erst durch den Ausgabeprozess in eventuell verschiedene physikalische Erscheinungsformen überführt wird. Jedes Ausgabegerät sollte dabei versuchen, die gewünschte graphische Struktur so genau physikalisch abzubilden, wie es seine prinzipiellen Fähigkeiten und sein momentaner technischer Zustand gestatten. Die Unterscheidung zwischen der graphischen Struktur und der physikalischen Erscheinungsform eines Dokuments ist analog der Unterscheidung zwischen einem Kreis als mathematischem Objekt und einem mit einem Zirkel auf ein Blatt Papier gezeichneten Kreis.

Die Beziehung zwischen graphischer Struktur, physikalischer und visueller Erscheinungsform eines Dokuments wird in Abbildung 1.1 veranschaulicht. In diesem Buch werden wir von diesen drei Ebenen fast nur die graphische betrachten.

Zu ihr gehören die Einteilung in Seiten, die Vergabe von Seitennummern, die räumliche Positionierung von Komponenten auf einer Seite, die Einteilung in Zeilen, horizontale und vertikale Abstände, Länge und Dicke von Linien sowie die Auswahl von Schriften nach Größe und Schriftart. In Kapitel 2 wird die graphische Dokumentstruktur genauer besprochen.

### 1.1.4 Die logische Struktur von Dokumenten

Über die an konkreten Erscheinungsformen zu erkennende graphische Struktur hinaus hat ein Dokument auch eine *logische Struktur*. Diese bezieht sich auf die vom Autor beabsichtigte Bedeutung der Dokumentbestandteile. Im Gegensatz dazu bezieht sich die graphische Form auf die vom Leser durch die physikalische Struktur erkennbare äußere Form der Dokumentbestandteile.

Dieses Buch z. B. besteht in seiner logischen Struktur aus Kapiteln, welche wiederum aus Abschnitten und Unterabschnitten bestehen. Jeder dieser Bestandteile besitzt eine eigene Überschrift. Graphisch/physikalisch spiegelt sich diese logische Struktur darin wider, daß ab und zu ein relativ großer Abstand zwischen zwei Zeilen zu beobachten ist, wobei jeweils nach diesem Abstand eine besonders groß und fett gedruckte Zeile mit einer bis drei Zahlen am Anfang folgt. Nahezu jeder Leser wird aus diesen Beobachtungen folgern, daß an dem großen Abstand ein Abschnitt oder Unterabschnitt endet und daß die fettgedruckte Zeile eine Überschrift ist, die den nächsten Abschnitt oder Unterabschnitt einleitet. Das würde auch funktionieren, wenn die Überschriften statt fett kursiv gedruckt oder unterstrichen wären. Wie an diesen Beispielen zu erkennen ist, gilt generell die Forderung: *Die graphische Struktur eines Dokuments soll die logische Struktur des Dokuments veranschaulichen.*

In der logischen Struktur eines Dokuments gibt es meist auf oberster Ebene eine Einteilung in einen Hauptteil und mehrere Nebenteile. Bei einem Buch können Inhalts-, Literatur- und Stichwortverzeichnis als Nebenteile auftreten. Bei einem Brief wären Anschrift, Absender, Betreff, Aktenzeichen, Datum und die Unterschrift mit der gewöhnlich davorstehenden Floskel als Nebenteile zu betrachten.

Der Hauptteil eines Dokuments besitzt oft eine hierarchische Struktur, die sich aus Kapiteln, Unterkapiteln, Abschnitten, Unterabschnitten und schließlich Absätzen (Paragraphen) aufbauen kann. Es hängt von der Art des Dokuments ab, wie viele solcher Hierarchieebenen tatsächlich verwirklicht sind.

Auf der Ebene der Absätze anzusiedeln sind Auflistungen mit Einträgen, die

- eigene Überschriften haben,
- durchnummeriert sind oder
- von besonderen Zeichen eingeleitet werden (wie im vorliegenden Fall).

Die Einträge in diesen Auflistungen können wieder aus mehreren Abschnitten bestehen sowie wiederum Auflistungen beinhalten.

Absätze oder Paragraphen bestehen logisch gesehen aus Sätzen und diese wiederum aus Wörtern. Auf der Wortebene können sich weitere logische Strukturen befinden, z. B. eingestreute Formeln wie  $E = mc^2$ , *Hervorhebungen*, Fußnoten<sup>1</sup>, Literaturzitate und Querverweise. Diese Strukturen können mitunter auch geschachtelt auftreten.

Querverweise sind besonders interessant, da sie an wohldefinierten Stellen innerhalb der hierarchischen logischen Struktur auftauchen, aber vollkommen hierarchieunabhängig auf beliebige andere Strukturen verweisen können. Es gibt logische Verweise, die sich auf logische Strukturelemente beziehen (z. B. „Kapitel 2“, „Satz 7.3“), und graphische Verweise, die auf Elemente der graphischen Struktur Bezug nehmen (z. B. „Seite 7“). Verweise werden in den Abschnitten 3.1.2, 3.2.2 und 3.3.3 besprochen.

Von der hierarchischen logischen Struktur des Textes sind Tabellen und Abbildungen oft unabhängig, nämlich wenn sie oben oder unten auf einer Seite, den normalen Textfluß unterbrechend, auftauchen oder gar eigene Seiten füllen.

Tabellen und Formeln besitzen normalerweise eine komplizierte innere logische Struktur. Tabellen bestehen aus Zeilen und Spalten, die sich in den einzelnen Tabellenfeldern kreuzen (siehe Kapitel 5). Formeln (siehe Kapitel 6) bestehen aus Teilformeln mit gewissen logischen Beziehungen, z. B. ein Bruch aus Zähler und Nenner (der Bruchstrich ist als Element der graphischen Struktur anzusehen).

Wie schon gesagt, sollen sich die logische Struktur eines Dokuments und die logischen Beziehungen zwischen seinen Bestandteilen in der graphischen Struktur des Dokuments ausdrücken. Die logischen Bestandteile müssen von ihrer Umgebung abgegrenzt werden und ihre gegenseitige Beziehung muß angedeutet werden. Möglichkeiten dazu sind das Einfügen horizontaler und vertikaler Zwischenräume oder eine besondere Positionierung, der Einsatz von typographischen Attributen wie Fettdruck oder Verkleinerung sowie die Benutzung von Hilfszeichen wie Strichen oder Nummern.<sup>2</sup>

Oft gibt es mehrere verschiedene Möglichkeiten, eine logische Struktur graphisch zu realisieren. Eine Hervorhebung im Text oder eine Überschrift könnte *kursiv*, **fett**, *g e s p e r r t* oder **vergrößert** geschrieben oder unterstrichen sein. Ein Bruch kann als  $\frac{3}{4}$  oder als  $3/4$  dargestellt werden. Andererseits setzen praktische Erwägungen und aus der Tradition geborene Konventionen gewisse Grenzen. Eine Hervorhebung, die genauso aussieht wie ihre Umgebung, ist nutzlos, und Gebilde wie  $3|4$  oder  $\frac{3}{4}$  wird niemand als Bruch erkennen.

---

1 wie z. B. diese

2 Fußnoten z. B. werden im allgemeinen durch eine besondere Positionierung ausgezeichnet, verkleinert geschrieben, durch einen Strich vom normalen Text abgegrenzt und mit Hilfe einer Nummer zu ihrer eigentlichen logischen Position im Text in Beziehung gesetzt.

## 1.1.5 Generische und spezifische Struktur

Wenn sich in einem Dokument gewisse Strukturelemente wiederholen, so redet man von *generischer* Struktur. Zur generischen graphischen Struktur gehören im allgemeinen die „normale“ Zeilenlänge, die Höhe des bedruckten Raums auf den Seiten, die Position der Seitennummer und das generelle Aussehen von Kapitelüberschriften (Schriftart und -größe). Zur generischen logischen Struktur können Aussagen gehören wie z. B. daß es Kapitel und Abschnitte gibt und daß ein Kapitel aus einer nichtleeren endlichen Folge von Abschnitten besteht.

Im Gegensatz zur generischen Struktur gibt die *spezifische* Struktur an, wie sich die generische Struktur in jedem einzelnen Falle realisiert. Obwohl es z. B. eine generische „normale“ Zeilenlänge gibt, sind die Zeilen am Ende eines Abschnitts meist kürzer. Oft gibt es auch vereinzelte längere Zeilen. Zur spezifischen logischen Struktur gehören Aussagen wie „Das siebte Kapitel besteht aus genau neun Abschnitten“.

Es ist auch möglich, daß verschiedene Dokumente dieselbe generische graphische oder logische Struktur besitzen. Das kann z. B. auf alle Bücher aus einer bestimmten Reihe eines Verlags oder auf alle Übungsscheine eines bestimmten Professors zutreffen. Man sagt, daß Dokumente übereinstimmender generischer Struktur zur selben *Dokumentklasse* gehören.

## 1.2 Erstellung von Dokumenten

### 1.2.1 Traditionelle Verfahren

Die einfachste Methode zur Dokumenterstellung ist wohl, das Dokument mit der Hand oder mit einer Schreibmaschine direkt auf Papier zu schreiben. Das Dokument wird sofort in der fertigen physikalischen Erscheinungsform erstellt. Eine logische oder graphische Struktur existiert allenfalls im Kopf des Autors. Sie entsteht meist erst inkrementell während des Schreibprozesses.

Im traditionellen Publikationsprozeß dagegen erzeugt der Autor erst einmal ein Handschriften- oder Schreibmaschinenmanuskript, das in etwa die beabsichtigte graphische Struktur aufweist. Dieses Manuskript wird an einen Verlag weitergereicht, wo es von einem Lektor mit zusätzlichen Markierungen *ausgezeichnet* wird, die Schriftarten bezeichnen sowie exakte numerische Werte für Abstände, Einrückungen und Schriftgrößen vorgeben. Der Lektor versucht also, die vom Autor geplante logische Struktur aus Inhalt und Aussehen des Manuskripts zu rekonstruieren und in eine passende graphische Struktur umzusetzen. Das ausgezeichnete Manuskript geht schließlich an einen Setzer, der unter Beachtung der Anweisungen des Lektors die endgültige graphische Struktur erzeugt. Diese wird durch eine Druckmaschine in eine physikalische Erscheinungsform des Dokuments umgesetzt.

## 1.2.2 Computersatz

Mit der Verbreitung von billigen und leistungsfähigen Rechnern wurde ein anderes Verfahren zur Dokumenterstellung üblich. Der Autor arbeitet dabei als sein eigener Lektor und erzeugt ein Manuskript mit Auszeichnungen. Dieses wird dann von einem dokumentverarbeitenden System, das die Rolle des Setzers übernimmt, in eine graphische Form umgesetzt, die von geeigneten *Ausgabegeräten* physikalisch dargestellt werden kann.

Bei älteren Systemen ist das Auszeichnen des Manuskripts graphisch orientiert, so wie es auch im Verlagswesen üblich ist. Der Autor gibt an, daß gewisse Textstücke fett oder kursiv gesetzt werden sollen oder daß numerisch genau spezifizierte horizontale oder vertikale Abstände eingefügt werden sollen. Das bringt Konsistenzprobleme mit sich, denn der Autor ist dann selbst dafür verantwortlich, daß z. B. alle Abschnittsüberschriften gleichartig gesetzt werden und den gleichen Abstand vom Folgetext haben.

Daher haben sich viele der Systeme mit graphisch orientierter Auszeichnung weiterentwickelt zu solchen, die auch logisch orientierte Auszeichnung unterstützen. Der Trick, der diese Weiterentwicklung ermöglicht hat, ist die Benutzung von *Makros*. Ein Makro ist ein Kommando, das als Abkürzung für eine längere Folge von Kommandos (Auszeichnungen) steht, die auch textuelle Parameter enthalten kann. Ein Überschriftsmakro z. B. könnte als Textparameter den Text der Überschrift enthalten und als Abkürzung für die folgenden Aktionen stehen: beende aktuellen Paragraph, erzeuge einen gewissen vertikalen Abstand, setze den Textparameter fett und in einer gewissen Schriftgröße, beende die aktuelle Zeile, erzeuge einen gewissen vertikalen Abstand. Komfortablere Überschriftenmakros können zusätzlich die Nummer des gerade beginnenden Abschnitts feststellen und ausgeben sowie dafür sorgen, daß die Überschrift ins Inhaltsverzeichnis aufgenommen wird und am Oberrand der folgenden Seiten erscheint.

Der Autor kann sich diese Makros im Prinzip selbst definieren, wird aber im allgemeinen auf vorgefertigte Makropakete zurückgreifen, die mit dem dokumentverarbeitenden System mitgeliefert werden. Diese Makropakete haben eine standardisierende Wirkung, da sie ja eine generische logische und graphische Struktur definieren.

Schließlich gibt es auch Systeme, die direkt im Hinblick auf eine logische Manuskriptauszeichnung hin konzipiert worden sind. Diese Systeme unterstützen im allgemeinen eine reichhaltigere logische Struktur als die makrobasierten Systeme. Meist bieten sie auch eine eigene Sprache zur Definition generischer logischer Strukturen an. Es wird dann automatisch getestet, ob die spezifische logische Struktur eines Dokuments zur angegebenen generischen Struktur kompatibel ist.

## 1.2.3 Batch-Systeme

Bezüglich der Handhabung von dokumentverarbeitenden Systemen lassen sich zwei große Klassen unterscheiden; *Batch-Systeme* und die im nächsten Unterab-

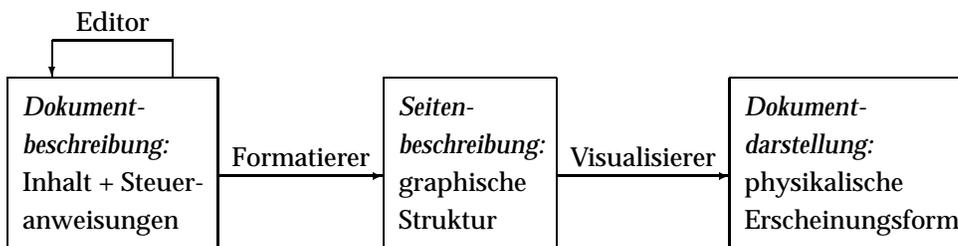


Abbildung 1.2: Arbeitsweise eines Batch-Systems

schnitt behandelten *interaktiven Systeme*. Natürlich gibt es auch Mischformen zwischen diesen beiden Klassen.

Bei der Benutzung eines Batch-Systems erzeugt der Autor zunächst eine Beschreibung seines Dokuments (Inhalt zusammen mit logischen und/oder graphischen Steueranweisungen) in der zum System gehörenden Dokumentbeschreibungssprache. Zur Erzeugung der Dokumentbeschreibung kann der Autor einen *Texteditor* seiner Wahl benutzen, der nicht unbedingt etwas mit dem benutzten Dokumentsystem zu tun haben muß. Der eigentliche Kern des Dokumentsystems ist der *Formatierer*. Er erzeugt aus der Dokumentbeschreibung eine graphische Form des Dokuments in einer Seitenbeschreibungssprache. Dazu muß er fortlaufenden Text in Zeilen und Seiten aufteilen, Seitennummern vergeben, Formeln und Tabellen setzen usw. Die Seitenbeschreibung kann dann durch einen *Visualisierer* auf einem *Ausgabegerät* (Drucker oder Bildschirm) als physikalische Dokumentdarstellung sichtbar gemacht werden (siehe Abbildung 1.2).

Um die Handhabung eines Batch-Systems näher zu erläutern, wird im folgenden anhand des Systems  $\text{\LaTeX}$  [Lam94] erklärt, wie reiner Text und die Beispielformel  $\frac{n+1}{n}$  erzeugt werden können.

Ein reiner Text wird mit beliebigen Zeilenwechseln und Wortabständen in die Dokumentbeschreibung eingetippt. Der Formatierer ignoriert diese vorgegebenen Zeilenwechsel und Abstände. Aus einer Folge von Wörtern erzeugt er Zeilen etwa gleicher Länge. In diesen werden die Wortzwischenräume etwas gedehnt oder gestaucht, um exakt gleiche Längen zu erhalten (Randausgleich).

Um die Formel  $\frac{n+1}{n}$  zu erzeugen, muß sie in der Dokumentbeschreibung spezifiziert werden. Eine Beschreibung der Formel in der Dokumentbeschreibungssprache des Systems  $\text{\LaTeX}$  sieht so aus: `\[ \frac{n+1}{n} \]`. Dabei bezeichnen `\[` und `\]` Anfang und Ende von Formeln, und `\frac{Zähler}{Nenner}` ist die  $\text{\LaTeX}$ -Beschreibung eines Bruchs. Es handelt sich um eine logische Beschreibung, da sie nichts über Länge und Dicke des Bruchstrichs aussagt.

Der  $\text{\LaTeX}$ -Formatierer erzeugt daraus einen Teil einer Seitenbeschreibung, die die genaue Form des Bruchstrichs angibt, die Position des Nenners in der Mitte darunter, die vertikalen Abstände von Zähler und Nenner zum Bruchstrich, sowie die

$n$	$\Rightarrow$	$n $
$+$	$\Rightarrow$	$n + \boxed{?}$
$1$	$\Rightarrow$	$n + 1 $
Leertaste	$\Rightarrow$	$n + \boxed{1}$
Leertaste	$\Rightarrow$	$\boxed{n + 1}$
$/$	$\Rightarrow$	$\frac{n+1}{\boxed{?}}$
$n$	$\Rightarrow$	$\frac{n+1}{n }$

Tabelle 1.1: Eingabe eines Bruches im FrameMaker

Anweisung, einen kleinen horizontalen Zwischenraum links und rechts von '+' einzufügen. Die Visualisierer erzeugen daraus die physikalische Darstellung  $\frac{n+1}{n}$  auf Papier oder Bildschirm.

Bei Benutzung eines Batch-Systems führt der Autor notwendige Änderungen normalerweise auf der Ebene der Dokumentbeschreibung durch und läßt dann wieder formatieren und darstellen. Die Seitenbeschreibung ist für Nichteingeweihte unverständlich, könnte im Prinzip aber auch mit einem Texteditor bearbeitet werden.

## 1.2.4 Interaktive Dokumentsysteme

Bei interaktiven Dokumentsystemen sind die im Falle der Batch-Systeme voneinander unabhängigen drei Hauptbestandteile Editor, Formatierer und Visualisierer in einem Programm zusammengefaßt. Der Autor sieht daher immer sein Dokument in weitgehend physikalischer Form auf dem Bildschirm und kann es durch geeignete Systemkommandos erweitern, verändern oder ausdrucken. Daher heißen diese Systeme auch *WYSIWYG-Systeme* nach den Anfangsbuchstaben des englischen „what you see is what you get“, d. h. was Sie sehen ist das, was Sie erhalten.

Die folgenden Beispiele zur Handhabung eines interaktiven Dokumentsystems beziehen sich auf das System *FrameMaker* [Fra93]. Ein reiner Text wird fortlaufend eingetippt. Immer wenn eine Zeile voll wird, erzeugt das System von selbst einen Zeilenwechsel und verändert die Zwischenräume in der gerade beendeten Zeile. Dies wird direkt am Bildschirm sichtbar.

Um eine Formel wie  $\frac{n+1}{n}$  einzugeben, muß der Autor dem System zunächst mitteilen, daß jetzt eine Formel beginnt. Dies kann durch Auswählen des Eintrags „New Medium Equation“ aus dem Menü „Equations“ geschehen. Danach kann die Eingabe der eigentlichen Formel erfolgen. In Tabelle 1.1 steht jeweils links die Eingabe und rechts das, was daraufhin auf dem Bildschirm zu sehen ist. Dabei bezeichnet der senkrechte Strich die Einfügestelle, d. h. die Stelle, an der das nächste eingegebene Zeichen erscheinen wird. Die Kästen bezeichnen selektierte Teilformeln,

d. h. Teilformeln, die für Folgeoperationen ausgezeichnet worden sind. Jeder Druck auf die Leertaste vergrößert die Selektion auf die nächstgrößere Teilformel. (Beim zweiten Druck auf die Leertaste im Beispiel oben vergrößert sich die Selektion von 1 auf  $n + 1$ , weil  $+ 1$  keine legale Teilformel ist.) Die Eingabe von '/' bewirkt die Bildung eines Bruchs, dessen Zähler die zu diesem Zeitpunkt selektierte Teilformel ist, während der Nenner noch undefiniert ist (?). Wenn der Autor im Beispiel oben die beiden Leertastendrücke vergißt, so erhält er statt  $\frac{n+1}{n}$  die Formel  $n + \frac{1}{n}$ . Nachdem die Formel vollständig eingegeben worden ist, wird dem System das Ende der Formel angezeigt durch Auswahl des Eintrags „Shrink-Wrap Equation“ aus dem Menü „Equations“.

## 1.2.5 Vergleich der Verfahren

Der Vorteil eines interaktiven Systems gegenüber einem Batchsystem ist, daß der Effekt jeder Eingabe oder Änderung auf das fertige Dokument sofort sichtbar wird. Daher können Fehler am Dokument unmittelbar nach ihrer Entdeckung korrigiert werden. Bei Batchsystemen dagegen sind die Eingabe/Änderung und das Betrachten des Dokuments zeitlich und räumlich getrennte Vorgänge. Wenn der Autor beim Betrachten des Dokuments einen Fehler sieht, so muß er erst die entsprechende Stelle in der Dokumentbeschreibung finden, um den Fehler korrigieren zu können. Nach einer Korrektur oder Änderung muß erst wieder der Formatierer laufen, bevor der Effekt der Änderung betrachtet werden kann.

Die den Batchsystemen eigene Distanz zwischen der Eingabe eines Dokuments und seiner Betrachtung am Bildschirm kann auch ihre Vorteile haben. In einem Batchsystem ist es weniger leicht möglich, durch einen Eingabefehler das gesamte Dokument zu zerstören. Der Autor wird nicht durch die bei einem interaktiven System ständig wechselnde Bildschirmansicht abgelenkt. Er kann eher logisch orientiert arbeiten. Da die direkte Bildschirmansicht fehlt, brauchen graphische Details nicht sofort festgelegt zu werden, sondern können erst nachträglich definiert oder genau eingestellt werden.

Ein anderer Vorteil von Batchsystemen liegt darin, daß ein beliebiger Editor benutzt werden kann, um die Dokumentbeschreibung zu erstellen. Der Autor kann also den Editor benutzen, mit dem er bereits vertraut ist, und beim Umsteigen von einem Batchsystem auf ein anderes, moderneres entfällt die Notwendigkeit, sich auch an einen neuen Editor zu gewöhnen. Bei interaktiven Systemen dagegen ist der Editor fest ins System integriert und nicht vom Autor frei wählbar. Er ist auch oft deutlich weniger mächtig als reine Editoren.

Die oben genannten Vor- und Nachteile von Batchsystemen gegenüber interaktiven werden von verschiedenen Benutzern verschieden bewertet. Die Bevorzugung eines bestimmten Dokumentsystems durch einen Benutzer scheint eher eine Frage der Gewöhnung zu sein als durch objektive Kriterien bestimmt.

Bisher pflegten Batchformatierer meist einen deutlich größeren Funktionsumfang als die in ein interaktives System eingebauten Formatierer zu haben, vor allem, was

logische und globale Eigenschaften des Dokuments angeht. Der Grund dafür ist, daß Batchformatierer das ganze Dokument auf einmal formatieren und daher einen globalen Überblick gewinnen können, während interaktive Formatierer immer nur den zuletzt eingegebenen oder geänderten Dokumentbestandteil bearbeiten. Die interaktiven Systeme haben in ihren neuesten Versionen allerdings stark aufgeholt und bieten inzwischen einen ähnlichen Funktionsumfang an wie Batchsysteme.

Generell ist eine gewisse Konvergenz zwischen den beiden Klassen von Systemen zu beobachten. Bei Batchsystemen war die Betrachtung des formatierten Dokuments am Bildschirm zunächst nicht möglich; es mußten erst geeignete Darstellungsprogramme entwickelt werden. Diese werden zwar immer komfortabler, erlauben aber im allgemeinen noch nicht, Veränderungen am Dokument durchzuführen. Zu manchen interaktiven Systemen gibt es menschenlesbare rein textuelle Formate, die den Dokumentbeschreibungen der Batchsysteme entsprechen. Es ist z. B. möglich, mit einem beliebigen Editor eine *FrameMaker*-Dokumentbeschreibung zu erzeugen und diese erst nachträglich von *FrameMaker* in eine graphische Form formatieren zu lassen. Aktuelle Versionen des *FrameMaker* bieten außerdem die Möglichkeit an, Formelteile in einer textuellen logischen Notation einzugeben, die von  $\text{\TeX}$  übernommen wurde.

## 1.2.6 Rechnerdarstellung der graphischen Struktur

Das Ergebnis der rechnergestützten Dokumenterstellung ist zunächst eine rechnerinterne Darstellung der graphischen Struktur des fertigen Dokuments, die dann von einem Ausgabegerät wie Drucker oder Bildschirmtreiber auf Papier oder Bildschirm ausgegeben wird. Diese rechnerinterne Darstellung kann unabhängig vom Ausgabegerät auf zwei Ebenen erfolgen:

1. durch Beschreibung der abstrakten Struktur der Seiten (page templates). Dies erfolgt durch *Seitenaufbausprachen*, z. B. im System ODA (Office Document Architecture) [App90]. Auch das interne Kastenmodell des Systems  $\text{\TeX}$  gehört in diese Ebene. In einer Seitenaufbausprache wird eine Seite typischerweise als eine Ansammlung hierarchisch ineinandergeschachtelter rechteckiger Kästen oder Rahmen dargestellt. Zu jedem der Kästen gehören Attribute, die seine Ausmaße und die Lage innerhalb des umfassenden Kastens beschreiben. Am unteren Ende der Hierarchie liegen Elementarkästen, die nicht mehr aus anderen Kästen zusammengesetzt sind, sondern graphische Grundobjekte wie Buchstaben oder Linien enthalten.
2. durch Angabe der Plazierung von graphischen Grundobjekten auf der Seite. Dazu dienen *Seitenbeschreibungssprachen*, z. B. *dvi* (siehe Abschnitt 9.3) und *PostScript* (siehe Abschnitt 9.5). Sie bestehen im wesentlichen aus Anweisungen, Zeichen aus einem bestimmten Schriftsatz in bestimmter Größe zu drucken, gemischt mit Anweisungen zur horizontalen und vertikalen Bewegung auf der Seite.

Wie oben erwähnt, sind diese Rechnerdarstellungen im wesentlichen geräteunabhängig. Um ein konkretes Ausgabegerät anzusprechen, muß diese geräteunabhängige Darstellung in eine auf dieses Gerät spezialisierte übersetzt werden. Diese Übersetzung kann auf dem Rechner selbst mit geeigneten Programmen ausgeführt werden. Es gibt aber auch Ausgabegeräte, die dafür programmiert sind, eine bestimmte geräteunabhängige Rechnerdarstellung zu verstehen und die Übersetzung in die für sie geeignete Spezialform selbst vorzunehmen. Um die von einem solchen Gerät verstandene Darstellung zu erreichen, ist es oft notwendig, von einer rechnerunabhängigen Darstellung in eine andere zu übersetzen. Beim System  $\text{T}_\text{E}_\text{X}$  [Knu86b] z. B. wird zunächst intern ein Seitenaufbau aus Kästen erzeugt, der dann in die Seitenbeschreibungssprache *dvi* übersetzt wird. Bei Bedarf kann die *dvi*-Beschreibung nach *PostScript* übersetzt werden.

### 1.2.7 Ausgabegeräte

Wie schon im vorigen Abschnitt beschrieben, soll bei der rechnergestützten Dokumenterstellung letztlich eine Darstellung des Dokuments auf einem Substrat durch ein Ausgabegerät erzeugt werden.

Bei einem *zeichenorientierten Ausgabegerät* ist das Ausgabemedium in (unsichtbare) rechteckige Kästchen gleicher Größe eingeteilt. In jedes Kästchen kann ein Zeichen aus einem beschränkten Zeichensatz eingefüllt werden. Beispiele sind herkömmliche Bildschirme und Typenraddrucker.

Ein *graphikorientiertes Ausgabegerät* kann nahezu beliebige Zeichen an beliebige Stellen des Ausgabemediums setzen. Graphikorientierte Ausgabegeräte sind typischerweise *Rasterausgabegeräte*, die ein Dokument als eine zweidimensionale Matrix von Punkten (genannt Pixel) ausgeben, z. B. Laser-, Nadel- und Tintenstrahldrucker sowie Rastergraphikbildschirme. Ein Schwarz-Weiß-Ausgabegerät färbt jeden Punkt entweder schwarz oder weiß (bzw. läßt ihn ungefärbt). Bei einem Graustufengerät können die Punkte mehrere verschiedene Graustufen zwischen schwarz und weiß annehmen. Auf einem Schwarz-Weiß-Gerät können Grautöne durch wechselnde Dichte der schwarzen Punkte hergestellt werden. Ein Farbausgabegerät ordnet jedem Punkt eine von einer (oft sehr großen) Anzahl von Farben zu.

Die *Auflösung* eines Rasterausgabegeräts ist die Anzahl von Punkten pro Längeneinheit in horizontaler wie vertikaler Richtung. Laserdrucker haben typischerweise eine Auflösung zwischen 300 und 600 Punkten pro Zoll, Fotosatzgeräte eine Auflösung von etwa 1000 Punkten pro Zoll. Bildschirme erreichen dagegen nur etwa 50 bis 150 Punkte pro Zoll.

Ein Schwarz-Weiß-Rasterausgabegerät kann am besten durch *Bitmatrizen* gesteuert werden. Eine Matrix ist eine zweidimensionale Anordnung von Einträgen, die im Falle von Bitmatrizen den Bildpunkten in ihrer zweidimensionalen Anordnung entsprechen. Jeder Eintrag ist ein *Bit*, also eine Ja-Nein-Information, die angibt, ob der zugehörige Punkt des Ausgabemediums geschwärzt werden soll.

Geräteunabhängige Beschreibungen der graphischen Dokumentstruktur müssen zur Ausgabe auf einem Schwarz-Weiß-Rasterausgabegerät in eine Bitmatrix konvertiert werden (scan conversion). Dabei müssen graphische Grundobjekte wie Buchstaben und Linien aus ihrer Darstellung im Speicher des Rechners in geeignete Gruppierungen von schwarzen Punkten auf weißem Untergrund übersetzt werden. Dazu muß die Gruppe von Punkten bestimmt werden, die das Objekt am besten auf dem Ausgabegerät wiedergibt. Da nicht unbedingt jede gewünschte Linienführung genau den Positionen der Punkte entspricht, müssen Sichtbarkeits- und Lesbarkeitsregeln bestimmen, welche Punkte in der Darstellung des Objekts jeweils schwarz bzw. weiß sein sollen. Die Pixeldarstellungen von Buchstaben der Standardschriften sind im allgemeinen bereits vorgegeben und müssen nicht für jeden Buchstaben neu berechnet werden.

Bitmatrizen sind natürlich geräteabhängig, weil die Beschreibung berücksichtigen muß, wie viele Bildpunkte das Ausgabegerät pro Flächeneinheit zu erzeugen vermag. Außerdem haben Bitmatrizen den Nachteil, daß sie sehr voluminös sind, da jeder einzelne Buchstabe schon aus vielen Bildpunkten besteht. Bei vielen Druckern geschieht die Übersetzung von einer geräteunabhängigen Form in Bitmatrizen daher erst im Inneren des Druckers, um die Verbindung zwischen Rechner und Drucker nicht zu sehr zu belasten. In diesem Falle versteht also der Drucker eine oder mehrere geräteunabhängige Darstellungen der graphischen Struktur.

## 1.3 Dienstprogramme zur Dokumentenverarbeitung

In diesem Abschnitt werden einige Dienstprogramme zur Dokumentenverarbeitung vorgestellt und anhand ihrer Funktionalität klassifiziert. Einige dieser Dienstprogramme (Editoren, Formatierer und interaktive Systeme) werden im späteren Verlauf dieses Buches noch genauer besprochen.

### Texteditoren

Texteditoren sind interaktive Werkzeuge zum Erstellen von (fast) unstrukturierten Texten. Diese Texte können auch Programme (Eingabe in Übersetzer) oder Dokumentbeschreibungen sein (Eingabe in Formatierer). Texte dieser Art sind natürlich doch strukturiert, aber der Texteditor weiß im allgemeinen fast nichts von dieser Struktur und kann insbesondere nicht zwischen Strukturierungsinformation und gewöhnlichem Text unterscheiden.

Mögliche Funktionen eines Editors sind: Herumwandern im Text; Einsetzen, Löschen, Ersetzen, Kopieren von Textteilen; Lesen und Schreiben von Dateien; globale Suche nach Wörtern sowie globale Ersetzungen. Texteditoren werden in Kapitel 8 genauer beschrieben.

## Formatierer

Formatierer im Batchbetrieb lesen eine Dokumentbeschreibung aus einer Datei und erzeugen daraus eine Seitenbeschreibung, die die graphische Struktur des formatierten Dokuments darstellt. Dabei werden unter anderem die folgenden Aufgaben bearbeitet:

1. Auswahl von Schriftarten und -größen (siehe Abschnitt 2.3)
2. Zeilenumbruch mit Randausgleich (Abschnitt 4.3)
3. Trennung von Wörtern am Zeilenende (Abschnitt 4.5)
4. Seitenumbruch (Abschnitt 4.4)
5. Setzen von Fußnoten, Tabellen (Kapitel 5), Formeln (Kapitel 6) usw.
6. automatische Numerierung von Komponenten wie z. B. Kapiteln, Definitionen, Abbildungen, Fußnoten und Aufzählungen (Abschnitte 3.2.1 und 3.3.1)
7. Behandlung von Verweisen (Abschnitte 3.1.2, 3.2.2 und 3.3.3)
8. Erstellen von abgeleiteten Dokumentteilen, z. B. Inhalts-, Stichwort- und Literaturverzeichnis
9. Erkennen von logischen Fehlern, z. B. zwei Formelanfänge ( $\backslash[$  in  $\LaTeX$ ) hintereinander ohne ein Formelende ( $\backslash]$ ) dazwischen
10. Verwalten von Variablen und benutzerdefinierten Kommandos, sogenannten *Makros* (Kapitel 7)

Formatierer können klassifiziert werden nach der Art der Dokumentbeschreibung, die sie verstehen. Die Beschreibung kann eher graphisch orientiert sein (*troff*,  $\TeX$ ) oder eher logisch orientiert (*Scribe*,  $\LaTeX$ ). Zwischen beiden Arten besteht ein fließender Übergang.

## Visualisierer (Previewer und Drucker)

Visualisierer lesen eine Seitenbeschreibung aus einer Datei und stellen das beschriebene Dokument physikalisch dar. Visualisierer sind einerseits als *Druckertreiber* in Drucker integriert, die eine Darstellung des Dokuments auf dem Papier liefern. Andererseits gibt es sogenannte *Previewer*, die eine Darstellung auf dem Bildschirm erzeugen, die möglichst genau der auf dem Papier entsprechen soll.

Zu den Funktionen, die von Previewern angeboten werden, gehören das Verschieben von Seiten auf dem Bildschirm, das Umblättern zur nächsten oder vorhergehenden Seite oder zu einer beliebigen anderen Seite, das Verkleinern des Dokuments zur Gewinnung eines Gesamtüberblicks, das Vergrößern von Dokumentteilen zum besseren Erkennen kleiner Details sowie das Drucken ausgewählter Seiten oder des ganzen Dokuments.

## Interaktive Textsysteme

Solche Systeme enthalten sowohl Editor- als auch Formatier- und Visualisierungsfunktionen. Bei älteren Systemen ist der Funktionsumfang dieser Komponenten geringer als der der entsprechenden „unabhängigen“ Dienstprogramme. Andererseits trifft man mitunter Funktionen, die bei reinen Formatierern unüblich sind, wie Rechtschreibprüfung, die Unterstützung von Serienbriefen oder die Möglichkeit, Einträge in Auflistungen automatisch sortieren zu lassen.

## Syntaxorientierte Editoren

Dies sind Editoren, die auf das Bearbeiten von Programmtexten einer festen Programmiersprache spezialisiert sind. Sie bieten typischerweise die folgenden Funktionen an:

- Syntaxüberprüfung von frei eingegebenem Programmtext
- Bereitstellen von Schablonen für Programmkonstrukte, die vom Programmierer ausgefüllt werden können
- Überprüfung von Kontextbedingungen (z. B. sind alle Variablen deklariert, die benutzt werden? Stimmen die Typen?)
- Einrückungen und Zeilenumbruch
- Drucken des Programms
- Ausführung des Programms

Syntaxorientierte Editoren für eine feste Programmiersprache können aus einer Beschreibung dieser Sprache automatisch erzeugt werden (z. B. durch den Cornell Program Synthesizer Generator [RT89]).

## Spezielle Formatierer

Gelegentlich trifft man spezielle Formatierer, die nur Tabellen oder nur Formeln bearbeiten können. Diese speziellen Formatierer werden als Präprozessoren zu allgemeineren Formatierern benutzt, die die betreffenden speziellen Fähigkeiten nicht anbieten. Beispiele sind die Bestandteile des *troff*-Systems (siehe Abschnitt 1.4.1).

## Graphiksysteme

Diese können zum Erstellen und Bearbeiten von Zeichnungen (mit Beschriftungen) benutzt werden. Es ist leider oft schwierig, fertige Zeichnungen in Textdokumente zu integrieren, da die entsprechenden Dateiformate oft nicht zueinander passen.

## Outliner

Outliner oder Dokument-Grobentwurfssysteme helfen beim Entwurf und der Organisation von strukturierten Dokumenten. Der Benutzer entwirft und verändert das Inhaltsverzeichnis bis zu beliebiger Tiefe.

## Desktop-Publishing-Systeme

Diese Systeme werden benutzt, um Dokumentbestandteile auf Seiten zweidimensional anzuordnen. Diese Bestandteile können von anderen Systemen vorformatierte Texte sein, Graphiken oder auch Photographien. Eine typische Anwendung ist das Erstellen von Zeitungsseiten aus Artikeln, Überschriften, Fotos, Werbeanzeigen usw.

## Hilfsprogramme

Vor allem für die englische Sprache sind Hilfsprogramme zur Rechtschreibprüfung und Stilbewertung verfügbar. Um ein allgemeines Programm zur Rechtschreibprüfung, das auf reinen Texten arbeitet, sinnvoll auf Dokumentbeschreibungen anwenden zu können, müssen diese erst von Anweisungen zur graphischen und/oder logischen Auszeichnung befreit werden. Auch dafür gibt es geeignete Hilfsprogramme. In manchen interaktiven Textsystemen wie z. B. *FrameMaker* ist die Rechtschreibprüfung direkt eingebaut.

## Übersetzer

Es gibt Programme, die von einer Seitenbeschreibungssprache in eine andere übersetzen. Wenn man z. B. den Formatierer  $\text{\LaTeX}$  besitzt, der Seitenbeschreibungen in der Sprache *dvi* erzeugt, und dazu einen Drucker, der nur Seitenbeschreibungen in *PostScript* versteht, dann hilft das Programm *dvips* [Rok95], das von *dvi* nach *PostScript* übersetzt.

Es gibt auch Programme, die Dokumentbeschreibungen für *troff* in solche für  $\text{\LaTeX}$  übersetzen.

## 1.4 Einige dokumentverarbeitende Systeme

In diesem Abschnitt werden einige der häufiger in diesem Buch erwähnten dokumentverarbeitenden Systeme vorgestellt.

### 1.4.1 *nroff* und *troff*

Zusammen mit dem Betriebssystem UNIX werden die Dokumentsysteme *nroff* und *troff* [Oss76] verteilt. Diese sind nahe miteinander verwandt; *nroff* formatiert ein Dokument zur Ausgabe auf zeichenorientierten Ausgabegeräten, z. B.

herkömmlichen Bildschirmen und Typenraddruckern, während *troff* in Hinblick auf Rasterausgabegeräte wie z. B. Graphikbildschirme oder Laserdrucker formatiert. In der folgenden Beschreibung konzentrieren wir uns auf *troff*.

Das *troff*-System ist ein Batch-System. Der Autor erzeugt also mit einem Texteditor seiner Wahl eine Dokumentbeschreibung in der *troff*-Eingabesprache, die dann zu einer Seitenbeschreibung formatiert wird. Die Aufgabe des Formatierens teilen sich drei verschiedene Programme: *eqn* [KC75] bearbeitet Formeln, *tbl* [Les76] setzt Tabellen und das eigentliche *troff*-Programm erledigt den Rest, d. h. die Bearbeitung von gewöhnlichem Text, Zeilenumbruch und Seitenumbruch.

Das Formatieren geschieht also in drei Schritten. Zunächst wird die vom Autor erzeugte Dokumentbeschreibung von *tbl* bearbeitet. Dieses Programm erzeugt eine neue Dokumentbeschreibung, in der die Beschreibungen von Tabellen durch primitivere graphische Formatieranweisungen ersetzt sind. Diese zweite Beschreibung wird von *eqn* bearbeitet, das die Beschreibung von Formeln durch primitivere Anweisungen ersetzt. Danach enthält die Dokumentbeschreibung nur noch Anweisungen, die für das eigentliche *troff*-Programm verständlich sind. Ein Dokument, das gar keine Tabellen enthält, muß nicht eigens vom Programm *tbl* bearbeitet werden. Ähnliches gilt für die Formeln.

Die in einer *troff*-Dokumentbeschreibung vorkommenden Anweisungen dienen vor allem zur graphischen Auszeichnung. Um sie von normalem Text zu unterscheiden, stehen sie in eigenen Zeilen, die mit einem Punkt eingeleitet werden. Gewisse Anweisungen, die durch das Symbol '\ ' eingeleitet werden, dürfen auch im Inneren gewöhnlicher Textzeilen vorkommen.

Einige Anweisungen beziehen sich auf eine gewisse Anzahl von Folgezeilen der Dokumentbeschreibung. Daher ist die Zeileneinteilung in der Dokumentbeschreibung nicht völlig irrelevant. Sie hat allerdings keinen direkten Einfluß auf die Zeileneinteilung im fertigen Dokument.

In *troff* ist es möglich, Makros zu definieren. Diese Möglichkeit ist dadurch eingeschränkt, daß Makronamen nur aus höchstens zwei Zeichen bestehen können. Mit dem *troff*-System werden einige Makropakete mitgeliefert, die eine mehr logisch orientierte Auszeichnung ermöglichen. Die Möglichkeiten, hierarchische logische Strukturen zu spezifizieren, sind aber begrenzt, weil das Konzept der hierarchischen Strukturierung der *troff*-Beschreibungssprache fremd ist. Eine Ausnahme stellen nur die Formeln dar, in deren Inneren beliebige Teilformeln durch die Klammern '{' und '}' abgegrenzt werden können.

## 1.4.2 $\TeX$ und $\LaTeX$

Das System  $\TeX$  [Knu86b, App88] von Donald E. Knuth ist wie *troff* ein Batch-System mit weitgehend graphisch orientierter Auszeichnung. Im Gegensatz zu *troff* sind aber alle Formatierfunktionen in einem einzigen Programm konzentriert, das gewöhnlichen Text, Formeln und Tabellen zu bearbeiten vermag. Nahezu alle

Anweisungen werden durch das Zeichen `\` eingeleitet und können an beliebiger Stelle in der Dokumentbeschreibung vorkommen. Außer der Tatsache, daß eine Leerzeile Abschnitte voneinander trennt, wird kein Bezug auf die Zeileneinteilung der Beschreibung genommen. Ein Hierarchiekonzept ist dadurch angelegt, daß beliebige Teile der Beschreibung durch die Klammern `{` und `}` zu einer Untereinheit zusammengefaßt werden können. Diese Klammern begrenzen im allgemeinen auch die Parameter von primitiven Anweisungen und Makros.

In  $\text{T}_{\text{E}}\text{X}$  ist es mit Leichtigkeit möglich, parametrisierte Makros zu definieren, deren Namen beliebig lang sein können. Das System  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  [Lam94, Kop96] von Leslie Lamport ist eine makrodefinierte Erweiterung von  $\text{T}_{\text{E}}\text{X}$ , die das Spezifizieren auch geschachtelter logischer Strukturen fördert. In begrenztem Rahmen wird die spezifizierte logische Struktur auf innere Konsistenz (z. B. ordentliche Klammerung) getestet.

Zu  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  gehören noch eine ganze Reihe von sogenannten Stildateien, das sind weitere Makropakete, die generische logische und graphische Strukturen beschreiben. Es ist allerdings nicht möglich, das genaue gegenseitige Verhältnis von logischen Strukturen anzugeben, z. B. daß ein Kapitel aus Abschnitten bestehen *muß*.

In diesem Buch wird  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  als das Hauptbeispiel eines Batch-Systems benutzt. Weitere Informationen zur Benutzung und Implementierung von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  und auch  $\text{T}_{\text{E}}\text{X}$  sind daher an vielen Stellen des Buchs zu finden.

### 1.4.3 SGML und DSSSL

Während *troff* und  $\text{T}_{\text{E}}\text{X}$  in erster Linie Formatierprogramme sind, die in zweiter Linie durch das, was sie als Eingabe erwarten, eine Sprache definieren, ist es mit SGML umgekehrt: SGML ist in erster Linie eine Sprache, zu der geeignete Implementierungen gefunden werden müssen.

Die Sprache SGML (Standard General Markup Language) [ISO86, GR90, Bry88] dient zur Definition von generischen logischen Strukturen und zur Spezifikation dazu passender spezifischer logischer Strukturen. Implementiert wird SGML durch einen SGML-Parser. Dieser hat die Aufgabe, zunächst eine generische Strukturdefinition zu lesen und dann ein konkretes Dokument mit spezifischer Struktur. Er muß dann testen, ob die angetroffene spezifische Struktur mit der definierten generischen übereinstimmt, und wenn ja, eine baumartige interne Darstellung des Dokuments in seiner (hierarchischen) spezifischen logischen Struktur erzeugen.

Ansonsten kann ein SGML-Parser nicht viel tun, denn in SGML kann man nichts anderes als den Inhalt und die logische Struktur von Dokumenten beschreiben. Es gibt keine Möglichkeit, eine graphische Struktur zu definieren. Diese Lücke wird von der Sprache DSSSL (Document Style Semantics and Specification Language) [ISO96] gefüllt. In DSSSL kann man generische graphische Strukturen (Seitenlayouts) definieren und festlegen, wie die in SGML definierte generische logische Struktur auf diese Seitenlayouts abgebildet wird.

Die Sprache SGML wird in Abschnitt 3.4 genauer besprochen. In Abschnitt 3.5 wird die verwandte Sprache HTML (Hypertext Markup Language) [Tol96] eingeführt, und DSSSL wird in Abschnitt 3.6 behandelt.

#### 1.4.4 ODA

Ähnlich wie SGML und DSSSL ist ODA (Office Document Architecture) [App90] zunächst nur ein Systemvorschlag, der erst realisiert werden muß. Im Gegensatz zu SGML und DSSSL werden in ODA die logische und die graphische Struktur zusammen definiert und behandelt. Ein ODA-Dokument in Internform besteht aus zwei an den Blättern verbundenen Bäumen. Ein Baum beschreibt die hierarchische logische Struktur, der andere die hierarchische graphische Struktur aus ineinandergeschachtelten Kästen. Die gemeinsamen Blätter der beiden Bäume beinhalten den eigentlichen Inhalt des Dokuments. Es ist möglich, sowohl die logische als auch die graphische Struktur zu bearbeiten. Die jeweils andere Struktur vollzieht die Veränderungen automatisch mit gemäß der spezifizierten generischen Beziehung, die zwischen den beiden Strukturen besteht.

#### 1.4.5 *FrameMaker*

*FrameMaker* [Fra93] ist ein interaktives Textsystem, das Edier-, Formatier- und Darstellungsfunktionen beinhaltet. Wie bereits erwähnt, ist es eine Charakteristik der interaktiven Systeme, daß das fertige Dokument immer gleich am Bildschirm erkennbar ist. Jede Aktion des Autors bewirkt eine sofortige Veränderung dieser Bildschirmsicht.

*FrameMaker* ist auf Workstations unter geeigneten Bildschirmoberflächen lauffähig. Er braucht nur einmal aufgerufen zu werden, was das Erscheinen eines Stammfensters bewirkt. Von diesem aus lassen sich Fenster abrufen, die einzelne Dokumente bearbeiten. Die Tatsache, daß mehrere Dokumente parallel in verschiedenen Fenstern in einem einzigen Programmablauf bearbeitet werden können, hat den Vorteil, daß sich Dokumentteile genauso leicht zwischen verschiedenen Dokumenten verschieben lassen wie innerhalb eines einzigen.

In einem Dokumentfenster ist ein Teil des Dokuments zu sehen, das gerade bearbeitet wird; es erscheint im wesentlichen so, wie es nach dem Drucken aussehen würde. Dieser Teil ist meist eine Seite, kann aber auch ein Teil einer Seite sein oder das untere Ende einer Seite zusammen mit dem Oberteil der nächsten; die Seiten sind dann durch einen fetten Strich getrennt.

Am Rande des Dokumentfensters befinden sich Menüköpfe, die beim Selektieren ein Menü bereitstellen, Kästchen zum Direktaufruf von Funktionen, Hilfsmittel zum horizontalen und vertikalen Verschieben des Sichtfensters auf dem Dokument sowie (wenn gewünscht) Lineale in einem selbstgewählten Maßstab. Unter anderem gibt es Kommandokästchen zum Vorwärts- und Rückwärtsblättern, zum

Vergrößern oder Verkleinern des Fensters (nicht des Dokuments) sowie zum Vergrößern oder Verkleinern des Dokuments (nicht des Fensters). Das Fenster kann mit Hilfe von Mausoperationen verschoben und manipuliert werden.

Wie gesagt, entspricht das Aussehen des Dokuments im Fenster im wesentlichen dem nach dem Drucken zu erwartenden Aussehen. Man kann allerdings auch wahlweise verschiedene logische Strukturen und graphische Hilfsinformationen mit anzeigen lassen, die dann nur auf dem Bildschirm zu sehen sind, aber nicht nach dem Druck. Dazu gehören Raster zur Ausrichtung, Kästen um logische Einheiten herum sowie spezielle Zeichen, die unter anderem das Ende von Abschnitten sowie vom Autor erzwungene oder verhinderte Zeilenumbrüche anzeigen.

Schon beim Aufruf kann man festlegen, welches die benutzte Sprache sein soll. Das bezieht sich nicht nur auf die Trennregeln und den eingebauten Rechtschreibtest, sondern auch auf die Meldungen und die Bezeichnungen in den Menüs, sogar auf die Tastaturcodes für Kommandos.

Die Bearbeitung des Dokuments erfolgt durch Benutzen von *Kommandos*. Kommandos können auf zwei verschiedene Weisen ausgelöst werden: durch Anwahl mit der Maus aus einem Menü (Problem: in welchem Menü ist welches Kommando versteckt?) oder durch Eingabe von zwei bis drei Zeichen, von denen das erste gewöhnlich ein Sonderzeichen ist (Problem: welche Zeichenfolge löst welches Kommando aus?).

Manche Kommandos können sofort ausgeführt werden, da sie nur einen Parameter haben, nämlich ein Textstück, das vorher mit der Maus selektiert werden muß. Kommandos, die noch andere Parameter benötigen, bewirken das Auftauchen spezieller Dialogfenster. Solche Dialogfenster werden auch benützt, um Voreinstellungen angeben zu können.

Je nach Art der Parameter und der möglichen Eingaben sind die Dialogfenster optisch verschieden gestaltet. Es gibt dabei drei grundlegende Möglichkeiten, die noch in mehreren Varianten vorkommen können:

1. Ein beliebiger Text kann frei eingegeben werden.
2. Aus einer festen Anzahl möglicher Eingaben wird mit der Maus genau eine ausgewählt.
3. Aus einer festen Anzahl möglicher Eingaben werden mit der Maus einige ausgewählt, deren Effekte sich dann kombinieren.

In diesem Buch wird *FrameMaker* (Version 4) als das Hauptbeispiel eines interaktiven Textsystems benutzt. Viele weitere Informationen zu seiner Handhabung und Funktionalität sind daher über das ganze Buch verteilt zu finden.