11 DOKUMENTE

11.1 DOKUMENTE

Die Idee zu dieser Einheit geht auf eine Vorlesung „Informatik A“ von Till Tantau aus dem Jahre 2012 zurück.

Im alltäglichen Leben hat man mit vielerlei Inschriften zu tun: Briefe, Kochrezepte, Zeitungsartikel, Vorlesungsskripte, Seiten im WWW, Emails, und so weiter. Bei vielem, was wir gerade aufgezählt haben, und bei vielem anderen kann man drei verschiedene Aspekte unterscheiden, die für den Leser eine Rolle spielen, nämlich

- den *Inhalt* des Textes, — *Inhalt*
- seine *Struktur* und — *Struktur*
- sein *Erscheinungsbild*, die (äußere) *Form*. — *Erscheinungsbild, Form*

Stünde die obige Aufzählung so auf dem Papier:

- den INHALT des Textes,
- seine STRUKTUR und
- sein ERSCHEINUNGSBILD, die (äußere) FORM.

dann hätte man an der Form etwas geändert (Wörter in Großbuchstaben statt kursiv), aber nicht am Inhalt oder an der Struktur. Hätten wir dagegen geschrieben:

> „[...] den *Inhalt* des Textes, seine *Struktur* und sein *Erscheinungsbild*, die (äußere) *Form*.“

dann wäre die Struktur eine andere (keine Liste mehr, sondern Fließtext), aber der Inhalt immer noch der gleiche. Und stünde hier etwas über

- *Balaenoptera musculus* (Blauwal),
- *Mesoplodon carlhubbsi* (Hubbs-Schnabelwal) und
- *Physeter macrocephalus* (Pottwal),

dann wäre offensichtlich der Inhalt ein anderer.

Von Ausnahmen abgesehen (welche fallen Ihnen ein?), ist Autoren und Lesern üblicherweise vor allem am Inhalt gelegen. Die Wahl einer bestimmten Struktur und Form hat primär zum Ziel, den Leser beim Verstehen des Inhalts zu unterstützen. Mit dem Begriff *Dokumente* in der Überschrift sollen Texte gemeint sein, bei denen man diese drei Aspekte unterscheiden kann. — *Dokumente*

Wenn Sie später beginnen, erste nicht mehr ganz kleine Dokumente (z. B. Seminarausarbeitungen oder die Bachelor-Arbeit) selbst zu schreiben, werden Sie merken, dass die Struktur, oder genauer gesagt das Auffinden einer geeigneten Struktur, auch zu Ihrem, also des Autors, Verständnis beitragen kann. Deswegen ist es bei solchen Arbeiten unseres Erachtens sehr ratsam, *früh damit zu beginnen,* — *ein Rat für Ihr weiteres Studium*

etwas aufzuschreiben, weil man so gezwungen wird, über die Struktur nachzudenken.

Programme fallen übrigens auch in die Kategorie dessen, was wir hier Dokumenten nennen.

11.2 Struktur von Dokumenten

Auszeichnungssprache
markup language

Oft ist es so, dass es Einschränkungen bei der erlaubten Struktur von Dokumenten gibt. Als Beispiel wollen wir uns zwei sogenannte *Auszeichnungssprachen* (im Englischen *markup language*) ansehen. Genauer gesagt werden wir uns dafür interessieren, wie Listen in LaTeX aufgeschrieben werden müssen, und wie Tabellen in XHTML.

11.2.1 LaTeX

LaTeX, ausgesprochen „Latech“, ist (etwas ungenau, aber für unsere Belange ausreichend formuliert) eine Erweiterung eines Textsatz-Programmes (iniTeX), das von Donald Knuth entwickelt wurde (http://www-cs-faculty.stanford.edu/~knuth/, 27.11.2015).

Es wird zum Beispiel in der Informatik sehr häufig für die Verfassung von wissenschaftlichen Arbeiten verwendet, weil unter anderem der Textsatz mathematischer Formeln durch TeX von hervorragender Qualität ist, ohne dass man dafür viel tun muss. Sie ist deutlich besser als alles, was der Autor dieser Zeilen jemals in Dokumenten gesehen hat, die mit Programmen wie z. B. libreoffice o. ä. verfasst wurden. Zum Beispiel liefert der Text

```
\[ 2 - \sum_{i=0}^{k} i 2^{-i} = (k+2) 2^{-k} \]
```

die Ausgabe

$$2 - \sum_{i=0}^{k} i2^{-i} = (k+2)2^{-k}$$

Auch der vorliegende Text wurde mit LaTeX gemacht. Für den Anfang dieses Abschnittes wurde z. B. geschrieben:

```
\section{Struktur von Dokumenten}
```

woraus LaTeX die Zeile

7.2 Struktur von Dokumenten

am Anfang dieser Seite gemacht hat. Vor dem Text der Überschrift wurde also automatisch die passende Nummer eingefügt und der Text wurde in einer Kapitälchenschrift gesetzt. Man beachte, dass z. B. die Auswahl der Schrift *nicht* in der

Eingabe mit vermerkt ist. Diese Angabe findet sich an anderer Stelle, und zwar an *einer* Stelle, an der das typografische Aussehen *aller* Abschnittsüberschriften (einheitlich) festgelegt ist.

Ganz grob kann ein LaTeX-Dokument z. B. die folgende Struktur haben:

```
\documentclass[11pt]{report}
% Kommentare fangen mit einem %-Zeichen an und gehen bis zum Zeilenende
% dieser Teil heißt Präambel des Dokumentes
% die folgende Zeile ist wichtig, kann hier aber nicht erklärt werden
\usepackage[T1]{fontenc}
% Unterstützung für Deutsch, z.B. richtige automatische Trennung
\usepackage[ngerman]{babel}
% Angabe des Zeichensatzes, der für den Text verwendet wird
\usepackage[utf8]{inputenc}     % zu UTF-8 siehe Kapitel 10
% für das Einbinden von Grafiken
\usepackage{graphicx}
\begin{document}
  % und hier kommt der eigentliche Text .....
\end{document}
```

Eine Liste einfacher Punkte sieht in LaTeX so aus:

Eingabe	Ausgabe
`\begin{itemize}`	
`\item Inhalt`	• Inhalt
`\item Struktur`	• Struktur
`\item Form`	• Form
`\end{itemize}`	

Vor den aufgezählten Wörtern steht jeweils ein dicker Punkt. Auch dieser Aspekt der äußeren Form ist *nicht* dort festgelegt, wo die Liste steht, sondern an *einer* Stelle, an der das typografische Aussehen *aller* solcher Listen (einheitlich) festgelegt ist. Wenn man in seinen Listen lieber alle Aufzählungspunkte mit einem sogenannten Spiegelstrich „–“ beginnen lassen möchte, dann muss man nur an einer Stelle (in der Präambel) die Definition von `\item` ändern.

Wollte man die formale Sprache L_{itemize} aller legalen Texte für Listen in LaTeX aufschreiben, dann könnte man z. B. zunächst geeignet die formale Sprache L_{item} aller Texte spezifizieren, die hinter einem Aufzählungspunkt vorkommen dürfen. Dann wäre

$$L_{\text{itemize}} = \{\texttt{\textbackslash begin\{itemize\}}\} \left(\{\texttt{\textbackslash item}\} L_{\text{item}}\right)^{+} \{\texttt{\textbackslash end\{itemize\}}\}$$

Dabei haben wir jetzt vereinfachend so getan, als wäre es kein Problem, L_{item} zu definieren. Tatsächlich ist das aber zumindest auf den ersten Blick eines, denn

ein Aufzählungspunkt in LaTeX darf seinerseits wieder eine Liste enthalten. Bei naivem Vorgehen würde man also genau umgekehrt für die Definition von L_{item} auch auf L_{itemize} zurückgreifen (wollen). Wir diskutieren das ein kleines bisschen ausführlicher in Unterabschnitt 11.2.3.

11.2.2 HTML und XHTML

HTML ist die Auszeichnungssprache, die man benutzt, wenn man eine WWW-Seite (be)schreibt. Für HTML ist formaler als für LaTeX festgelegt, wie syntaktisch korrekte solche Seiten aussehen. Das geschieht in einer sogenannten *document type definition*, kurz *DTD*.

Hier ist ein Auszug aus der DTD für eine (nicht die allerneueste) Version von XHTML. Sie dürfen sich vereinfachend vorstellen, dass das ist im wesentlichen eine noch striktere Variante von HTML ist. Das nachfolgenden Fragment beschreibt teilweise, wie man syntaktisch korrekt eine Tabelle notiert.

```
<!ELEMENT table (caption?, thead?, tfoot?, (tbody+|tr+))>
<!ELEMENT caption  %Inline;>
<!ELEMENT thead    (tr)+>
<!ELEMENT tfoot    (tr)+>
<!ELEMENT tbody    (tr)+>
<!ELEMENT tr       (th|td)+>
<!ELEMENT th       %Flow;>
<!ELEMENT td       %Flow;>
```

Wir können hier natürlich nicht auf Details eingehen. Einige einfache aber wichtige Aspekte sind aber mit unserem Wissen schon verstehbar. Die Wörter wie `table`, `thead`, `tr`, usw. dürfen wir als bequem notierte Namen für formale Sprachen auffassen. Welche, das wollen wir nun klären.

Die Bedeutung von `*` und `+` ist genau das, was wir als Konkatenationsabschluss und ε-freien Konkatenationsabschluss kennen gelernt haben. Die Bedeutung des Kommas `,` in der ersten Zeile ist die des Produktes formaler Sprachen. Die Bedeutung des senkrechten Striches `|` in der sechsten Zeile ist Vereinigung von Mengen.

Das Fragezeichen ist uns neu, hat aber eine ganz banale Bedeutung: In der uns geläufigen Notation würden wir definieren: $L^? = L^0 \cup L^1 = \{\varepsilon\} \cup L$. Mit anderen Worten: Wenn irgendwo $L^?$ notiert ist, dann kann an dieser Stelle ein Wort aus L stehen, oder es kann fehlen. Das Auftreten eines Wortes aus L ist also mit anderen Worten optional.

Nun können Sie zur Kenntnis nehmen, dass z. B. die Schreibweise

```
<!ELEMENT tbody (tr)+ >
```

die folgende formale Sprache festlegt:

$$L_{\texttt{tbody}} = \{\texttt{<tbody>}\} \cdot L_{\texttt{tr}}^{+} \cdot \{\texttt{</tbody>}\}$$

Das heißt, ein Tabellenrumpf (*table body*) beginnt mit der Zeichenfolge <tbody>, endet mit der Zeichenfolge </tbody>, und enthält dazwischen eine beliebige positive Anzahl von Tabellenzeilen (*table rows*). Und die erste Zeile aus der DTD besagt

$$L_{\texttt{table}} = \{\texttt{<table>}\} \cdot L_{\texttt{caption}}^{?} \cdot L_{\texttt{thead}}^{?} \cdot L_{\texttt{tfoot}}^{?} \cdot (L_{\texttt{tbody}}^{+} \cup L_{\texttt{tr}}^{+}) \cdot \{\texttt{</table>}\}$$

das heißt, eine Tabelle (*table*) ist von den Zeichenketten <table> und </table> umschlossen und enthält innerhalb in dieser Reihenfolge

- optional eine Überschrift (*caption*),
- optional einen Tabellenkopf (*table head*),
- optional einen Tabellenfuß (*table foot*) und
- eine beliebige positive Anzahl von Tabellenrümpfen (siehe eben) oder Tabellenzeilen.

Insgesamt ergibt sich, dass zum Beispiel

```
<table>
  <tbody>
    <tr>  <td>1</td> <td>a</td>  </tr>
    <tr>  <td>2</td> <td>b</td>  </tr>
  </tbody>
</table>
```

eine syntaktisch korrekte Tabelle.

In Wirklichkeit gibt es noch zusätzliche Aspekte, die das Ganze formal verkomplizieren und bei der Benutzung flexibler machen, aber das ganz Wesentliche haben wir damit jedenfalls an einem Beispiel beleuchtet.

11.2.3 Eine Grenze unserer bisherigen Vorgehensweise

Dass wir eben mit Hilfe von Produkt und Konkatenationsabschluss formaler Sprachen in einigen Fällen präzise Aussagen machen konnten, hing auch mit der Einfachheit dessen zusammen, was es zu spezifizieren galt. Es wurde, jedenfalls in einem intuitiven Sinne, immer etwas von einer komplizierteren Art aus Bestandteilen einfacherer Art zusammengesetzt.

Es gibt aber auch den Fall, dass man sozusagen größere Dinge einer Art aus kleineren Bestandteilen zusammensetzen will, die aber von der gleichen Art sind.

Auf Listen, deren Aufzählungspunkte ihrerseits wieder Listen enthalten dürfen, hatten wir im Zusammenhang mit LaTeX schon hingewiesen.

Ein anderes typisches Beispiel sind korrekt geklammerte arithmetische Ausdrücke. Sehen wir einmal von den Operanden und Operatoren ab und konzentrieren uns auf die reinen Klammerungen. Bei einer syntaktisch korrekten Klammerung gibt es zu jeder Klammer auf „weiter hinten“ die „zugehörige“ Klammer zu. Insbesondere gilt:

- Man kann beliebig viele korrekte Klammerungen konkatenieren und erhält wieder eine korrekte Klammerung.
- Man kann um eine korrekte Klammerung außen herum noch ein Klammerpaar schreiben (Klammer auf ganz vorne, Klammer zu ganz hinten) und erhält wieder eine korrekte Klammerung.

Man würde also gerne in irgendeinem Sinne L_{Klammer} mit L^*_{Klammer} und mit $\{()\cdot L_{\text{Klammer}}\cdot\{()\}$ in Beziehung setzen. Es ist aber nicht klar wie. Insbesondere würden bei dem Versuch, eine Art Gleichung hinzuschreiben, sofort die Fragen im Raum stehen, ob die Gleichung überhaupt lösbar ist, und wenn ja, ob die Lösung eindeutig ist.

11.3 ZUSAMMENFASSUNG

In dieser Einheit haben wir über *Dokumente* gesprochen. Sie haben einen *Inhalt*, eine *Struktur* und ein *Erscheinungsbild*.

Formale Sprachen kann man z. B. benutzen, um zu spezifizieren, welche Struktur(en) ein legales, d. h. syntaktisch korrektes Dokument haben darf, sofern die Strukturen hinreichend einfach sind. Was man in komplizierten Fällen machen kann, werden wir in einer späteren Einheit kennenlernen.

