Grundbegriffe der Informatik

Einheit 7: Dokumente

Thomas Worsch

Universität Karlsruhe, Fakultät für Informatik

November 2008

Überblick

Dokumente

- LaTeX
- XHTML
- Eine Grenze unserer bisherigen Vorgehensweise

Dokumente

- im alltäglichen Leben vielerlei Inschriften:
  Briefe, Kochrezepte, Zeitungsartikel, Vorlesungsskripte, Seiten im WWW, Emails, usw.
- oft drei verschiedene Aspekte unterscheidbar,
  die für den Leser eine Rolle spielen:
  - den Inhalt des Textes,
  - seine Struktur und
  - sein Erscheinungsbild, die (äußere) Form.

Inhalt, Struktur und Form

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

Inhalt, Struktur und Form

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

andere Form:

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

Inhalt, Struktur und Form

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

andere Struktur:

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

Inhalt, Struktur und Form

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

anderer Inhalt:

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

Wozu Inhalt, Struktur und Form?

- Inhalt üblicherweise für Autoren und Leser im Vordergrund
  - (Ausnahmen?)
- Struktur und Form
  - sollen den Leser beim Verstehen des Inhalts zu unterstützen.

Wozu Inhalt, Struktur und Form?

- Inhalt üblicherweise für Autoren und Leser im Vordergrund
  - (Ausnahmen?)
- Struktur und Form
  - sollen den Leser beim Verstehen des Inhalts zu unterstützen.

- Wir wollen als Dokumente solche Texte bezeichnen, bei denen man die drei Aspekte Inhalt, Struktur und Form unterscheiden kann.
- Programme gehören übrigens auch dazu.

Wozu Inhalt, Struktur und Form?

- Inhalt üblicherweise für Autoren und Leser im Vordergrund
  - (Ausnahmen?)
- Struktur und Form
  - sollen den Leser beim Verstehen des Inhalts zu unterstützen.

- Wir wollen als Dokumente solche Texte bezeichnen, bei denen man die drei Aspekte Inhalt, Struktur und Form unterscheiden kann.
- Programme gehören übrigens auch dazu.

- ein Rat für Ihr weiteres Studium:
  - Wenn Sie beginnen, erste nicht mehr ganz kleine Dokumente (Seminarausarbeitungen, Bachelor-Arbeit) zu schreiben,
  - werden Sie merken, dass die Struktur, genauer das Auffinden geeigneter Struktur, auch zu Ihrem Verständnis beiträgt.
  - *Deswegen ist es bei solchen Arbeiten ratsam früh damit zu beginnen, etwas aufzuschreiben,*
    (weil man gezwungen wird, über die Struktur nachzudenken).

Struktur von Dokumenten

- auch da spielt syntaktische Korrektheit eine Rolle,
- zumindest wenn Rechner im Spiel sind.
- Beispiele: Auszeichnungssprachen (engl. *markup language*)
  - Listen in LaTeX
    - und ein klein bisschen Allgemeines zu LaTeX
  - Tabellen in XHTML

- basiert auf TeX von Donald Knuth
  `http://www-cs-faculty.stanford.edu/~knuth/`
- ausgesprochen „Tech“ (bzw. „Latech“)
- Textsatz-Programm
- in der Informatik sehr häufig verwendet wegen
  - hervorragendem automatischen Satz mathematischer Formeln
  - aus
    ```
    \[ 2 - \sum_{i=0}^{k} i 2^{-i} = (k+2) 2^{-k} \]
    ```
  - wird

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

- Vorlesungsskript und Folien sind auch mit LaTeX gemacht
  - Diese Folie beginnt so:

```
\begin{frame}[fragile]
  \frametitle{\LaTeX}

  \begin{itemize}
  \item basiert auf \TeX{} von Donald Knuth \
```

- Im Skript steht zum Beispiel
  `\subsection{Struktur von Dokumenten}`
- woraus LaTeX die Zeile

  7.2 Struktur von Dokumenten

  am Anfang von Seite 40 des Skriptes gemacht hat
- also
  - automatisch die passende Abschnittsnummer eingefügt
  - Text wurde in einer Kapitälchenschrift gesetzt
- Beachte: z. B. Schriftauswahl ist *nicht* in der Eingabe mit vermerkt ist.
- Diese Festlegung findet sich an anderer Stelle, und zwar an *einer* Stelle, an der das typografische Aussehen *aller* Abschnittsüberschriften (einheitlich) festgelegt ist.

Grobstruktur von LaTeX-Dokumenten

```
  \documentclass[11pt]{report}
  % so schreibt man Kommentare
  % dieser Teil heißt Präambel des Dokumentes
  % Unterstützung für Deutsch,
  % z.B. richtige automatische Trennung
    \usepackage[german]{babel}
    % Angabe des Zeichensatzes, in dem der Text ist
    \usepackage[latin1]{inputenc}
    % 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}`	

- der dicker Punkt als Markierung ist *nicht* an der Stelle der Liste festgelegt.
- Trennung der Spezifikation von Struktur und der Spezifikation von Form
- Wenn man z. B. Spiegelstriche „–“ möchte, dann muss man an *einer* Stelle (in der Präambel) die Definition `\item` ändern.

Formale Sprachen

- Gesucht: die formale Sprache L_{itemize} aller legalen Texte für Listen in LaTeX
- Gegeben: die formale Sprache L_{item} aller Texte, die hinter einem Aufzählungspunkt vorkommen dürfen.
  (tun wir mal so . . .)
- Dann

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

- Problem: In LaTeX darf in einem Aufzählungspunkt wieder eine Liste vorkommen.
- Bei naivem Vorgehen würde man also auch umgekehrt für die Definition von L_{item} auf L_{itemize} zurückgreifen (wollen).

XHTML

- HTML: Auszeichnungssprache, die man benutzt, wenn man eine WWW-Seite (be)schreibt.
- XHTML: sozusagen im wesentlichen eine noch striktere Variante von HTML
- Für beide formaler als für LaTeX festgelegt, wie syntaktisch korrekte solche Seiten aussehen.
- Das geschieht in einer sogenannten document type definition, kurz *DTD*.

Auszug aus der DTD für Tabellen in XHTML

```
<!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;>
```

- hier: kaum Details
- das Wichtigste können wir schon verstehen

Interpretation der DTD für Tabellen in XHTML

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

- Wörter wie `table`, `thead`, `tr`, fassen wir als Namen für formale Sprachen auf.
- Bedeutung des `+` ist ε-freier Konkatenationsabschluss
- Bedeutung des Kommas `,` ist Produkt formaler Sprachen
- Bedeutung des senkrechten Striches `|` ist Vereinigung
- Fragezeichen ist neu, aber ganz einfach:

$$L^? = L^0 \cup L^1 = \{\varepsilon\} \cup L$$

- Wenn $L^?$ notiert, dann kann an dieser Stelle ein Wort aus L stehen, oder es kann fehlen.
  optionales Auftreten eines Wortes aus L

Interpretation der DTD für Tabellen in XHTML

- Mitteilung:
  `<!ELEMENT tbody (tr)+ >` legt fest:

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

- Tabellenrumpf beginnt mit Zeichenfolge `<tbody>`, endet mit `</tbody>`, und enthält dazwischen eine beliebige positive Anzahl von Tabellenzeilen
- erste Zeile aus der DTD besagt:

$$L_{\texttt{table}} = \{\texttt{<table>}\} \cdot L_{\texttt{caption}}^{?} \cdot L_{\texttt{thead}}^{?} \cdot L_{\texttt{tf}\infty\texttt{t}}^{?} \cdot L_{\texttt{tbody}}^{+} \cdot \{\texttt{</table>}\}$$

- Tabelle ist von 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.

Beispiel für Tabelle in XHTML

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

Eine Grenze unserer bisherigen Vorgehensweise

- eben mit Hilfe von Produkt und Konkatenationsabschluss formaler Sprachen präzise Aussagen gemacht.
- Das ging, weil
  - etwas von einer komplizierteren Art aus Bestandteilen einfacherer Art zusammengesetzt wurde.

Eine Grenze unserer bisherigen Vorgehensweise

- eben mit Hilfe von Produkt und Konkatenationsabschluss formaler Sprachen präzise Aussagen gemacht.
- Das ging, weil
  - etwas von einer komplizierteren Art aus Bestandteilen einfacherer Art zusammengesetzt wurde.
- Aber manchmal: größere Dinge einer Art werden zusammengesetzt aus kleineren Bestandteilen *der gleichen Art*
- Beispiel: Listen, deren Aufzählungspunkte ihrerseits wieder Listen enthalten dürfen ...

Eine Grenze unserer bisherigen Vorgehensweise (2)

- anderes typisches Beispiel:
  korrekt geklammerte arithmetische Ausdrücke
- 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 kan 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
  L_{Klammer} mit L^*_{Klammer} und mit $\{(\} \cdot L_{\text{Klammer}} \cdot \{)\}$
  in Beziehung setzen.
- Aber wie?
- Gleichung?
  - Ist sie lösbar?
  - wenn ja: Lösung eindeutig?

Zusammenfassung

- Dokumente haben Inhalt, Struktur und Form.
- formale Sprachen helfen bei der Definition legaler Strukturen.