

- ▶ Uniform Resource Locator (einheitlicher Quellenanzeiger)
 - ▶ Abkürzung „URL“ - ist ein AKRONYM
 - ▶ Aussprache: Lautkette („NASA“) oder Einzellaute („U S A“)
Acronym: USA: IPA: /ɜ:l/ Homophones: earl USA: IPA: /'ju:ɑ:r,ɛl/
- ▶ Warum kann man Wörter mit dt. Umlauten wie „örl“ in emails schreiben obwohl nur Zeichen aus dem ASCII-Alphabet erlaubt sind?
 - ▶ Gute Frage: Details in der Vorlesung „Codierung“
 - ▶ Kurzwantwort: Übersetzung mittels punycode (puny - mickrig)
- ▶ RFC 2822 wurde im Jahr 2008 durch RFC 5322 ersetzt
 - ▶ RFC 5322 <http://tools.ietf.org/html/rfc5322>
 - ▶ „This specification is a revision of RFC 2822, which itself superseded RFC 822, updating it to reflect current practice and incorporating incremental changes that were specified in other RFCs.“

- ▶ Uniform Resource Locator (einheitlicher Quellenanzeiger)
 - ▶ Abkürzung „URL“ - ist ein AKRONYM
 - ▶ Aussprache: Lautkette („NASA“) oder Einzellaute („U S A“)
Acronym: USA: IPA: /ɜ:l/ Homophones: earl USA: IPA: /'ju:ɑ:r,ɛl/
- ▶ Warum kann man Wörter mit dt. Umlauten wie „örl“ in emails schreiben obwohl nur Zeichen aus dem ASCII-Alphabet erlaubt sind?
 - ▶ Gute Frage: Details in der Vorlesung „Codierung“
 - ▶ Kurzwantwort: Übersetzung mittels punycode (puny - mickrig)
- ▶ RFC 2822 wurde im Jahr 2008 durch RFC 5322 ersetzt
 - ▶ RFC 5322 <http://tools.ietf.org/html/rfc5322>
 - ▶ „This specification is a revision of RFC 2822, which itself superseded RFC 822, updating it to reflect current practice and incorporating incremental changes that were specified in other RFCs.“

- ▶ Uniform Resource Locator (einheitlicher Quellenanzeiger)
 - ▶ Abkürzung „URL“ - ist ein AKRONYM
 - ▶ Aussprache: Lautkette („NASA“) oder Einzellaute („U S A“)
Acronym: USA: IPA: /ɜ:l/ Homophones: earl USA: IPA: /'ju:ɑ:r,ɛl/
- ▶ Warum kann man Wörter mit dt. Umlauten wie „örl“ in emails schreiben obwohl nur Zeichen aus dem ASCII-Alphabet erlaubt sind?
 - ▶ Gute Frage: Details in der Vorlesung „Codierung“
 - ▶ Kurzwantwort: Übersetzung mittels punycode (puny - mickrig)
- ▶ RFC 2822 wurde im Jahr 2008 durch RFC 5322 ersetzt
 - ▶ RFC 5322 <http://tools.ietf.org/html/rfc5322>
 - ▶ „This specification is a revision of RFC 2822, which itself superseded RFC 822, updating it to reflect current practice and incorporating incremental changes that were specified in other RFCs.“

- ▶ Uniform Resource Locator (einheitlicher Quellenanzeiger)
 - ▶ Abkürzung „URL“ - ist ein AKRONYM
 - ▶ Aussprache: Lautkette („NASA“) oder Einzellaute („U S A“)
Acronym: USA: IPA: /ɜ:l/ Homophones: earl USA: IPA: /'ju:ɑ:r,ɛl/
- ▶ Warum kann man Wörter mit dt. Umlauten wie „örl“ in emails schreiben obwohl nur Zeichen aus dem ASCII-Alphabet erlaubt sind?
 - ▶ Gute Frage: Details in der Vorlesung „Codierung“
 - ▶ Kurzantwort: Übersetzung mittels punycode (puny - mickrig)
- ▶ RFC 2822 wurde im Jahr 2008 durch RFC 5322 ersetzt
 - ▶ RFC 5322 <http://tools.ietf.org/html/rfc5322>
 - ▶ „This specification is a revision of RFC 2822, which itself superseded RFC 822, updating it to reflect current practice and incorporating incremental changes that were specified in other RFCs.“

- ▶ Uniform Resource Locator (einheitlicher Quellenanzeiger)
 - ▶ Abkürzung „URL“ - ist ein AKRONYM
 - ▶ Aussprache: Lautkette („NASA“) oder Einzellaute („U S A“)
Acronym: USA: IPA: /ɜ:l/ Homophones: earl USA: IPA: /'ju:ɑ:r,ɛl/
- ▶ Warum kann man Wörter mit dt. Umlauten wie „örl“ in emails schreiben obwohl nur Zeichen aus dem ASCII-Alphabet erlaubt sind?
 - ▶ Gute Frage: Details in der Vorlesung „Codierung“
 - ▶ Kurzantwort: Übersetzung mittels punycode (puny - mickrig)
- ▶ RFC 2822 wurde im Jahr 2008 durch RFC 5322 ersetzt
 - ▶ RFC 5322 <http://tools.ietf.org/html/rfc5322>
 - ▶ „This specification is a revision of RFC 2822, which itself superseded RFC 822, updating it to reflect current practice and incorporating incremental changes that were specified in other RFCs.“

- ▶ Uniform Resource Locator (einheitlicher Quellenanzeiger)
 - ▶ Abkürzung „URL“ - ist ein AKRONYM
 - ▶ Aussprache: Lautkette („NASA“) oder Einzellaute („U S A“)
Acronym: USA: IPA: /ɜ:l/ Homophones: earl USA: IPA: /'ju:ɑ:r,ɛl/
- ▶ Warum kann man Wörter mit dt. Umlauten wie „örl“ in emails schreiben obwohl nur Zeichen aus dem ASCII-Alphabet erlaubt sind?
 - ▶ Gute Frage: Details in der Vorlesung „Codierung“
 - ▶ Kurzwantwort: Übersetzung mittels punycode (puny - mickrig)
- ▶ RFC 2822 wurde im Jahr 2008 durch RFC 5322 ersetzt
 - ▶ RFC 5322 <http://tools.ietf.org/html/rfc5322>
 - ▶ „This specification is a revision of RFC 2822, which itself superseded RFC 822, updating it to reflect current practice and incorporating incremental changes that were specified in other RFCs.“

- ▶ Uniform Resource Locator (einheitlicher Quellenanzeiger)
 - ▶ Abkürzung „URL“ - ist ein AKRONYM
 - ▶ Aussprache: Lautkette („NASA“) oder Einzellaute („U S A“)
Acronym: USA: IPA: /ɜ:l/ Homophones: earl USA: IPA: /'ju:ɑ:r,ɛl/
- ▶ Warum kann man Wörter mit dt. Umlauten wie „örl“ in emails schreiben obwohl nur Zeichen aus dem ASCII-Alphabet erlaubt sind?
 - ▶ Gute Frage: Details in der Vorlesung „Codierung“
 - ▶ Kurzwantwort: Übersetzung mittels punycode (puny - mickrig)
- ▶ RFC 2822 wurde im Jahr 2008 durch RFC 5322 ersetzt
 - ▶ RFC 5322 <http://tools.ietf.org/html/rfc5322>
 - ▶ „This specification is a revision of RFC 2822, which itself superseded RFC 822, updating it to reflect current practice and incorporating incremental changes that were specified in other RFCs.“

- ▶ Uniform Resource Locator (einheitlicher Quellenanzeiger)
 - ▶ Abkürzung „URL“ - ist ein AKRONYM
 - ▶ Aussprache: Lautkette („NASA“) oder Einzellaute („U S A“)
Acronym: USA: IPA: /ɜ:l/ Homophones: earl USA: IPA: /'ju:ɑ:r,ɛl/
- ▶ Warum kann man Wörter mit dt. Umlauten wie „örl“ in emails schreiben obwohl nur Zeichen aus dem ASCII-Alphabet erlaubt sind?
 - ▶ Gute Frage: Details in der Vorlesung „Codierung“
 - ▶ Kurzwantwort: Übersetzung mittels punycode (puny - mickrig)
- ▶ RFC 2822 wurde im Jahr 2008 durch RFC 5322 ersetzt
 - ▶ RFC 5322 <http://tools.ietf.org/html/rfc5322>
 - ▶ „This specification is a revision of RFC 2822, which itself superseded RFC 822, updating it to reflect current practice and incorporating incremental changes that were specified in other RFCs.“

- ▶ Uniform Resource Locator (einheitlicher Quellenanzeiger)
 - ▶ Abkürzung „URL“ - ist ein AKRONYM
 - ▶ Aussprache: Lautkette („NASA“) oder Einzellaute („U S A“)
Acronym: USA: IPA: /ɜ:l/ Homophones: earl USA: IPA: /'ju:ɑ:r,ɛl/
- ▶ Warum kann man Wörter mit dt. Umlauten wie „örl“ in emails schreiben obwohl nur Zeichen aus dem ASCII-Alphabet erlaubt sind?
 - ▶ Gute Frage: Details in der Vorlesung „Codierung“
 - ▶ Kurzwantwort: Übersetzung mittels punycode (puny - mickrig)
- ▶ RFC 2822 wurde im Jahr 2008 durch RFC 5322 ersetzt
 - ▶ RFC 5322 <http://tools.ietf.org/html/rfc5322>
 - ▶ „This specification is a revision of RFC 2822, which itself superseded RFC 822, updating it to reflect current practice and incorporating incremental changes that were specified in other RFCs.“

Grundbegriffe der Informatik

Einheit 5: formale Sprachen

Prof. Dr. Tanja Schultz

Karlsruher Institut für Technologie, Fakultät für Informatik

Wintersemester 2011/2012

Formale Sprachen

- Formale Sprachen

- Produkt formaler Sprachen

- Konkatenationsabschluss formaler Sprachen

- ▶ Natürliche Sprachen sind gekennzeichnet durch:
 - ▶ Aussprache
 - ▶ Stil, z. B. Wortwahl und Satzbau
 - ▶ Welche Formulierungen sind syntaktisch korrekt?
 - ▶ Ist und syntaktisch welcher welcher Satz korrekt nicht?
- ▶ Informatik:
 - ▶ Sprachen, die nicht natürlich sind:
 - ▶ Programmiersprachen
 - ▶ Aufbau von Emails, WWW-Seiten, ...
 - ▶ Eingabedateien für ...
 - ▶ **Syntax**
 - ▶ Wie spezifiziert man, was korrekt ist?
 - ▶ Wie überprüft man, ob etwas korrekt ist?
 - ▶ **Semantik**
 - ▶ Wie definiert man, was syntaktisch korrekte Gebilde bedeuten?
 - ▶ Darum kümmern wir uns später

Formale Sprachen

Formale Sprachen

Produkt formaler Sprachen

Konkatenationsabschluss formaler Sprachen

- ▶ Alphabet A gegeben
- ▶ Eine **formale Sprache** (über einem Alphabet A) ist eine Teilmenge $L \subseteq A^*$.
- ▶ Im Zusammenhang mit syntaktischer Korrektheit:
 - ▶ formale Sprache L der syntaktisch korrekten Gebilde
 - ▶ syntaktisch falsche Gebilde gehören *nicht* zu L

Beispiele:

- ▶ $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -\}$.
formale Sprache der Dezimaldarstellungen ganzer Zahlen
 - ▶ enthält z.B. 1 , -22 und 192837465 ,
 - ▶ enthält aber nicht $2-3---41$.
- ▶ formale Sprache der syntaktisch korrekten Java-Programme
 - ▶ enthält alle Java-Programme
 - ▶ enthält aber zum Beispiel *nicht*: `[2] class int)(`

Formale Sprachen

Formale Sprachen

Produkt formaler Sprachen

Konkatenationsabschluss formaler Sprachen

- ▶ Wir kennen schon die Konkatination von Wörtern
- ▶ Wir definieren das Produkt der Sprachen L_1 und L_2 :

$$L_1 \cdot L_2 = \{w_1 w_2 \mid w_1 \in L_1 \wedge w_2 \in L_2\}$$

- ▶ Wegen der Assoziativität der Konkatination von Wörtern $(w_1 \cdot w_2) \cdot w_3 = w_1 \cdot (w_2 \cdot w_3)$ gilt auch:

$$L_1 \cdot L_2 \cdot L_3 = \{w_1 w_2 w_3 \mid w_1 \in L_1 \wedge w_2 \in L_2 \wedge w_3 \in L_3\}$$

usw.

- ▶ den Punkt „ \cdot “ lässt man auch gerne wieder weg

- ▶ Wir kennen schon die Konkatination von Wörtern
- ▶ Wir definieren das Produkt der Sprachen L_1 und L_2 :

$$L_1 \cdot L_2 = \{w_1 w_2 \mid w_1 \in L_1 \wedge w_2 \in L_2\}$$

- ▶ Wegen der Assoziativität der Konkatination von Wörtern $(w_1 \cdot w_2) \cdot w_3 = w_1 \cdot (w_2 \cdot w_3)$ gilt auch:

$$L_1 \cdot L_2 \cdot L_3 = \{w_1 w_2 w_3 \mid w_1 \in L_1 \wedge w_2 \in L_2 \wedge w_3 \in L_3\}$$

usw.

- ▶ den Punkt „ \cdot “ lässt man auch gerne wieder weg

- ▶ wenn $L_1 = \{a, aa\}$ und $L_2 = \{b, ab\}$
dann $L_1 \cdot L_2 = \{a, aa\} \cdot \{b, ab\}$
$$= \{a \cdot b, a \cdot ab, aa \cdot b, aa \cdot ab\}$$
$$= \{ab, aab, aaab\}$$
- ▶ wenn $S = \{\text{int}, \text{double}, \text{char}, \dots\}$, $B = \{a, \dots, z\}$ und $Z = \{0, \dots, 9\}$
dann enthält $S \cdot \{_ \} \cdot B \cdot (B \cup Z)^* \cdot \{; \}$
„Deklarationen“ wie z. B.
 - ▶ `int_x42;`
 - ▶ `double_wurzelzwei;`aber leider auch
 - ▶ `int_double;`andererseits aber nicht
 - ▶ `char__hugo_;`

Lemma. Für jede formale Sprache L ist

$$L \cdot \{\varepsilon\} = L = \{\varepsilon\} \cdot L.$$

Beweis durch einfaches Nachrechnen:

$$\begin{aligned} L \cdot \{\varepsilon\} &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 \in \{\varepsilon\}\} \\ &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 = \varepsilon\} \\ &= \{w_1 \varepsilon \mid w_1 \in L\} \\ &= \{w_1 \mid w_1 \in L\} \\ &= L \end{aligned}$$

Analog zeigt man $L = \{\varepsilon\} \cdot L$.

$\{\varepsilon\}$ ist das neutrale Element bzgl. des Produkts formaler Sprachen.

Lemma. Für jede formale Sprache L ist

$$L \cdot \{\varepsilon\} = L = \{\varepsilon\} \cdot L.$$

Beweis durch einfaches Nachrechnen:

$$\begin{aligned} L \cdot \{\varepsilon\} &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 \in \{\varepsilon\}\} \\ &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 = \varepsilon\} \\ &= \{w_1 \varepsilon \mid w_1 \in L\} \\ &= \{w_1 \mid w_1 \in L\} \\ &= L \end{aligned}$$

Analog zeigt man $L = \{\varepsilon\} \cdot L$.

$\{\varepsilon\}$ ist das neutrale Element bzgl. des Produkts formaler Sprachen.

Lemma. Für jede formale Sprache L ist

$$L \cdot \{\varepsilon\} = L = \{\varepsilon\} \cdot L.$$

Beweis durch einfaches Nachrechnen:

$$\begin{aligned} L \cdot \{\varepsilon\} &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 \in \{\varepsilon\}\} \\ &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 = \varepsilon\} \\ &= \{w_1 \varepsilon \mid w_1 \in L\} \\ &= \{w_1 \mid w_1 \in L\} \\ &= L \end{aligned}$$

Analog zeigt man $L = \{\varepsilon\} \cdot L$.

$\{\varepsilon\}$ ist das neutrale Element bzgl. des Produkts formaler Sprachen.

Lemma. Für jede formale Sprache L ist

$$L \cdot \{\varepsilon\} = L = \{\varepsilon\} \cdot L.$$

Beweis durch einfaches Nachrechnen:

$$\begin{aligned} L \cdot \{\varepsilon\} &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 \in \{\varepsilon\}\} \\ &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 = \varepsilon\} \\ &= \{w_1 \varepsilon \mid w_1 \in L\} \\ &= \{w_1 \mid w_1 \in L\} \\ &= L \end{aligned}$$

Analog zeigt man $L = \{\varepsilon\} \cdot L$.

$\{\varepsilon\}$ ist das neutrale Element bzgl. des Produkts formaler Sprachen.

Lemma. Für jede formale Sprache L ist

$$L \cdot \{\varepsilon\} = L = \{\varepsilon\} \cdot L.$$

Beweis durch einfaches Nachrechnen:

$$\begin{aligned} L \cdot \{\varepsilon\} &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 \in \{\varepsilon\}\} \\ &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 = \varepsilon\} \\ &= \{w_1 \varepsilon \mid w_1 \in L\} \\ &= \{w_1 \mid w_1 \in L\} \\ &= L \end{aligned}$$

Analog zeigt man $L = \{\varepsilon\} \cdot L$.

$\{\varepsilon\}$ ist das neutrale Element bzgl. des Produkts formaler Sprachen.

Lemma. Für jede formale Sprache L ist

$$L \cdot \{\varepsilon\} = L = \{\varepsilon\} \cdot L.$$

Beweis durch einfaches Nachrechnen:

$$\begin{aligned} L \cdot \{\varepsilon\} &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 \in \{\varepsilon\}\} \\ &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 = \varepsilon\} \\ &= \{w_1 \varepsilon \mid w_1 \in L\} \\ &= \{w_1 \mid w_1 \in L\} \\ &= L \end{aligned}$$

Analog zeigt man $L = \{\varepsilon\} \cdot L$.

$\{\varepsilon\}$ ist das neutrale Element bzgl. des Produkts formaler Sprachen.

Lemma. Für jede formale Sprache L ist

$$L \cdot \{\varepsilon\} = L = \{\varepsilon\} \cdot L.$$

Beweis durch einfaches Nachrechnen:

$$\begin{aligned} L \cdot \{\varepsilon\} &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 \in \{\varepsilon\}\} \\ &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 = \varepsilon\} \\ &= \{w_1 \varepsilon \mid w_1 \in L\} \\ &= \{w_1 \mid w_1 \in L\} \\ &= L \end{aligned}$$

Analog zeigt man $L = \{\varepsilon\} \cdot L$.

$\{\varepsilon\}$ ist das neutrale Element bzgl. des Produkts formaler Sprachen.

- ▶ wir wollen Potenzen L^k formaler Sprachen definieren
- ▶ „Problem“: Was soll L^0 sein?
- ▶ Definiere:

$$L^0 = \{\varepsilon\}$$

$$\forall k \in \mathbb{N}_0 : L^{k+1} = L \cdot L^k$$

- ▶ Einfaches Nachrechnen ergibt z. B.:

$$L^1 = L$$

$$L^2 = L \cdot L$$

$$L^3 = L \cdot L \cdot L$$

- ▶ Genau genommen: $L^3 = L \cdot (L \cdot L)$, aber:
Konkatenation von Sprachen ist eine assoziative Operation.

- ▶ wir wollen Potenzen L^k formaler Sprachen definieren
- ▶ „Problem“: Was soll L^0 sein?
- ▶ Definiere:

$$L^0 = \{\varepsilon\}$$

$$\forall k \in \mathbb{N}_0 : L^{k+1} = L \cdot L^k$$

- ▶ Einfaches Nachrechnen ergibt z. B.:

$$L^1 = L$$

$$L^2 = L \cdot L$$

$$L^3 = L \cdot L \cdot L$$

- ▶ Genau genommen: $L^3 = L \cdot (L \cdot L)$, aber:
Konkatenation von Sprachen ist eine assoziative Operation.

- ▶ wir wollen Potenzen L^k formaler Sprachen definieren
- ▶ „Problem“: Was soll L^0 sein?
- ▶ Definiere:

$$L^0 = \{\varepsilon\}$$

$$\forall k \in \mathbb{N}_0 : L^{k+1} = L \cdot L^k$$

- ▶ Einfaches Nachrechnen ergibt z. B.:

$$L^1 = L$$

$$L^2 = L \cdot L$$

$$L^3 = L \cdot L \cdot L$$

- ▶ Genau genommen: $L^3 = L \cdot (L \cdot L)$, aber:
Konkatenation von Sprachen ist eine assoziative Operation.

- ▶ wir wollen Potenzen L^k formaler Sprachen definieren
- ▶ „Problem“: Was soll L^0 sein?
- ▶ Definiere:

$$L^0 = \{\varepsilon\}$$

$$\forall k \in \mathbb{N}_0 : L^{k+1} = L \cdot L^k$$

- ▶ Einfaches Nachrechnen ergibt z. B.:

$$L^1 = L$$

$$L^2 = L \cdot L$$

$$L^3 = L \cdot L \cdot L$$

- ▶ Genau genommen: $L^3 = L \cdot (L \cdot L)$, aber:
Konkatenation von Sprachen ist eine assoziative Operation.

► $L = \{aa, b\}$

► Dann ist

$$L^0 = \{\varepsilon\}$$

$$L^1 = \{aa, b\}$$

$$\begin{aligned} L^2 &= \{aa, b\} \cdot \{aa, b\} = \{aa \cdot aa, aa \cdot b, b \cdot aa, b \cdot b\} \\ &= \{aaaa, aab, baa, bb\} \end{aligned}$$

$$\begin{aligned} L^3 &= \{aa \cdot aa \cdot aa, aa \cdot aa \cdot b, aa \cdot b \cdot aa, aa \cdot b \cdot b, \\ &\quad b \cdot aa \cdot aa, b \cdot aa \cdot b, b \cdot b \cdot aa, b \cdot b \cdot b\} \\ &= \{aaaaaa, aaaab, aabaa, aabb, baaaa, baab, bbaa, bbb\} \end{aligned}$$

- Sei

$$L = \{a^n b^n \mid n \in \mathbb{N}_+\},$$

also sozusagen (immer diese Pünktchen ...)

$$L = \{ab, aabb, aaabbb, aaaabbbb, \dots\}.$$

- Was ist $L^2 = L \cdot L$?



$$\begin{aligned} L^2 = & \{ab \cdot ab, ab \cdot aabb, ab \cdot aaabbb, \dots\} \\ & \cup \{aabb \cdot ab, aabb \cdot aabb, aabb \cdot aaabbb, \dots\} \\ & \cup \{aaabbb \cdot ab, aaabbb \cdot aabb, aaabbb \cdot aaabbb, \dots\} \\ & \vdots \end{aligned}$$

- Mit anderen Worten

$$L^2 = \{a^{n_1} b^{n_1} a^{n_2} b^{n_2} \mid n_1 \in \mathbb{N}_+ \wedge n_2 \in \mathbb{N}_+\}.$$

- Beachte: die Exponenten n_1 „vorne“ und n_2 „hinten“ heißen verschieden.

- ▶ Sei

$$L = \{a^n b^n \mid n \in \mathbb{N}_+\},$$

also sozusagen (immer diese Pünktchen ...)

$$L = \{ab, aabb, aaabbb, aaaabbbb, \dots\}.$$

- ▶ Was ist $L^2 = L \cdot L$?



$$\begin{aligned} L^2 = & \{ab \cdot ab, ab \cdot aabb, ab \cdot aaabbb, \dots\} \\ & \cup \{aabb \cdot ab, aabb \cdot aabb, aabb \cdot aaabbb, \dots\} \\ & \cup \{aaabbb \cdot ab, aaabbb \cdot aabb, aaabbb \cdot aaabbb, \dots\} \\ & \vdots \end{aligned}$$

- ▶ Mit anderen Worten

$$L^2 = \{a^{n_1} b^{n_1} a^{n_2} b^{n_2} \mid n_1 \in \mathbb{N}_+ \wedge n_2 \in \mathbb{N}_+\}.$$

- ▶ Beachte: die Exponenten n_1 „vorne“ und n_2 „hinten“ heißen verschieden.

- ▶ für Alphabet A und für $i \in \mathbb{N}_0$ hatten wir schon Potenzen A^i definiert.
- ▶ A^n : *Menge aller Wörter der Länge n* über dem Alphabet A
- ▶ Jedes Alphabet A kann man als formale Sprache L_A auffassen (enthält alle Wörter der Länge 1)
- ▶ Man mache sich klar:
 A^i ist („im Wesentlichen“) das Gleiche wie L_A^i .
 - ▶ $A^0 = \{\varepsilon\} = L_A^0$
 - ▶ $A^1 = A = L_A = L_A^1$
 - ▶ $A^2 = A \cdot A^1 = L_A \cdot L_A^1$
 - ▶ ...
 - ▶ wonach sieht das aus?

Formale Sprachen

Formale Sprachen

Produkt formaler Sprachen

Konkatenationsabschluss formaler Sprachen

- ▶ bei Alphabeten schon gesehen: $A^* = \bigcup_{i=0}^{\infty} A^i$.

- ▶ der **Konkatenationsabschluss** L^* von L ist

$$L^* = \bigcup_{i=0}^{\infty} L^i$$

- ▶ der **ε -freie Konkatenationsabschluss** L^+ von L ist

$$L^+ = \bigcup_{i=1}^{\infty} L^i$$

- ▶ Man sieht:

$$L^* = L^0 \cup L^+.$$

- ▶ $L = \{ab, c\}$
- ▶ $L^0 = \{\varepsilon\}$
- ▶ $L^1 = \{ab, c\}$
- ▶ $L^2 = \{abab, abc, cab, cc\}$
- ▶ $L^3 = \{ababab, ababc, abcab, abcc, cabab, cabcc, ccab, ccc\}$
- ▶ ...
- ▶ Der Konkatenationsabschluss von L ist $L^* = \bigcup_{i=0}^{\infty} L^i$
- ▶ $L^* = L^0 \cup L^1 \cup L^2 \cup \dots = \{\varepsilon, ab, c, abab, abc, cab, cc, \dots\}$
- ▶ Der ε -freie Konkatenationsabschluss von L ist $L^+ = \bigcup_{i=1}^{\infty} L^i$
- ▶ $L^+ = L^1 \cup L^2 \cup \dots = \{ab, c, abab, abc, cab, cc, \dots\}$
- ▶ Zusammenhang: $L^* = L^0 \cup L^+$

- **Abgeschlossenheit:** Ist f eine n -stellige innere Verknüpfung (hier binäre/zweistellige Operation „Konkatenation“) auf einer Menge A , dann heißt das: f ist eine Funktion $A^n \rightarrow A$.
Gilt nun $\emptyset \neq M \subseteq A$, dann heißt M abgeschlossen bezüglich f , wenn $f(a_1, \dots, a_n)$ in M liegt für alle $a_1, \dots, a_n \in M$, wenn also f eingeschränkt auf den Definitionsbereich M^n auch eine n -stellige innere Verknüpfung (hier Konkatenation) auf M ist.

- ▶ Es sei wieder $L = \{a^n b^n \mid n \in \mathbb{N}_+\}$.
- ▶ haben schon gesehen:

$$L^2 = \{a^{n_1} b^{n_1} a^{n_2} b^{n_2} \mid n_1 \in \mathbb{N}_+ \wedge n_2 \in \mathbb{N}_+\}.$$

- ▶ analog

$$L^3 = \{a^{n_1} b^{n_1} a^{n_2} b^{n_2} a^{n_3} b^{n_3} \mid n_1 \in \mathbb{N}_+ \wedge n_2 \in \mathbb{N}_+ \wedge n_3 \in \mathbb{N}_+\}.$$

- ▶ wir erlauben uns Pünktchen ...:

$$L^i = \{a^{n_1} b^{n_1} \dots a^{n_i} b^{n_i} \mid n_1, \dots, n_i \in \mathbb{N}_+\}.$$

- ▶ Dann kann man für L^+ notieren:

$$L^+ = \{a^{n_1} b^{n_1} \dots a^{n_i} b^{n_i} \mid i \in \mathbb{N}_+ \wedge n_1, \dots, n_i \in \mathbb{N}_+\}.$$

- ▶ Sie merken (hoffentlich): L^+ und L^* sind *präziser und kürzer* hinzuschreiben als viele Pünktchen.

- ▶ $A = \{a, b\}$
- ▶ $L = \{a\}^* \cup \{b\}^* = \{\epsilon, a, aa, aaa, \dots, b, bb, bbb, \dots\}$
- ▶ Welche Wörter enthält L^* ?
 - ▶ alle Wörter, die man erhält, wenn man
 - ▶ eine beliebige endliche Zahl k
 - ▶ von Wörtern w_1, \dots, w_k aus L
 - ▶ konkateniert zu $w_1 \cdot \dots \cdot w_k$
 - ▶ z. B. $aa \cdot \epsilon \cdot aaaa \cdot b \cdot aaaaa$
 - ▶ z. B. $aa \cdot bbbbbb \cdot aaa \cdot b \cdot aaaaa \cdot bbb \cdot aaa$
 - ▶ jedes Wort aus A^* ist die Konkatenation von Blöcken, die nur aus a oder nur aus b bestehen
 - ▶ also ist $L^* = A^*$

- ▶ $A = \{a, b\}$
- ▶ $L = \{a\}^* \cup \{b\}^* = \{\varepsilon, a, aa, aaa, \dots, b, bb, bbb, \dots\}$
- ▶ Welche Wörter enthält L^* ?
 - ▶ alle Wörter, die man erhält, wenn man
 - ▶ eine beliebige endliche Zahl k
 - ▶ von Wörtern w_1, \dots, w_k aus L
 - ▶ konkateniert zu $w_1 \cdot \dots \cdot w_k$
 - ▶ z. B. $aa \cdot \varepsilon \cdot aaaa \cdot b \cdot aaaaa$
 - ▶ z. B. $aa \cdot bbbbbb \cdot aaa \cdot b \cdot aaaaa \cdot bbb \cdot aaa$
 - ▶ jedes Wort aus A^* ist die Konkatenation von Blöcken, die nur aus a oder nur aus b bestehen
 - ▶ also ist $L^* = A^*$

- ▶ $A = \{a, b\}$
- ▶ $L = \{a\}^* \cup \{b\}^* = \{\varepsilon, a, aa, aaa, \dots, b, bb, bbb, \dots\}$
- ▶ Welche Wörter enthält L^* ?
 - ▶ alle Wörter, die man erhält, wenn man
 - ▶ eine beliebige endliche Zahl k
 - ▶ von Wörtern w_1, \dots, w_k aus L
 - ▶ konkateniert zu $w_1 \cdots w_k$
 - ▶ z. B. $aa \cdot \varepsilon \cdot aaaa \cdot b \cdot aaaaa$
 - ▶ z. B. $aa \cdot bbbbb \cdot aaa \cdot b \cdot aaaaa \cdot bbb \cdot aaa$
 - ▶ jedes Wort aus A^* ist die Konkatenation von Blöcken, die nur aus a oder nur aus b bestehen
 - ▶ also ist $L^* = A^*$

- ▶ $A = \{a, b\}$
- ▶ $L = \{a\}^* \cup \{b\}^* = \{\varepsilon, a, aa, aaa, \dots, b, bb, bbb, \dots\}$
- ▶ Welche Wörter enthält L^* ?
 - ▶ alle Wörter, die man erhält, wenn man
 - ▶ eine beliebige endliche Zahl k
 - ▶ von Wörtern w_1, \dots, w_k aus L
 - ▶ konkateniert zu $w_1 \cdots w_k$
 - ▶ z. B. $aa \cdot \varepsilon \cdot aaaa \cdot b \cdot aaaaa$
 - ▶ z. B. $aa \cdot bbbbb \cdot aaa \cdot b \cdot aaaaa \cdot bbb \cdot aaa$
 - ▶ jedes Wort aus A^* ist die Konkatenation von Blöcken, die nur aus a oder nur aus b bestehen
 - ▶ also ist $L^* = A^*$

- ▶ $A = \{a, b\}$
- ▶ $L = \{a\}^* \cup \{b\}^* = \{\varepsilon, a, aa, aaa, \dots, b, bb, bbb, \dots\}$
- ▶ Welche Wörter enthält L^* ?
 - ▶ alle Wörter, die man erhält, wenn man
 - ▶ eine beliebige endliche Zahl k
 - ▶ von Wörtern w_1, \dots, w_k aus L
 - ▶ konkateniert zu $w_1 \cdots w_k$
 - ▶ z. B. $aa \cdot \varepsilon \cdot aaaa \cdot b \cdot aaaaa$
 - ▶ z. B. $aa \cdot bbbbb \cdot aaa \cdot b \cdot aaaaa \cdot bbb \cdot aaa$
 - ▶ jedes Wort aus A^* ist die Konkatenation von Blöcken, die nur aus a oder nur aus b bestehen
 - ▶ also ist $L^* = A^*$

- ▶ $A = \{a, b\}$
- ▶ $L = \{a\}^* \cup \{b\}^* = \{\varepsilon, a, aa, aaa, \dots, b, bb, bbb, \dots\}$
- ▶ Welche Wörter enthält L^* ?
 - ▶ alle Wörter, die man erhält, wenn man
 - ▶ eine beliebige endliche Zahl k
 - ▶ von Wörtern w_1, \dots, w_k aus L
 - ▶ konkateniert zu $w_1 \cdots w_k$
 - ▶ z. B. $aa \cdot \varepsilon \cdot aaaa \cdot b \cdot aaaaa$
 - ▶ z. B. $aa \cdot bbbbb \cdot aaa \cdot b \cdot aaaaa \cdot bbb \cdot aaa$
 - ▶ jedes Wort aus A^* ist die Konkatenation von Blöcken, die nur aus a oder nur aus b bestehen
 - ▶ also ist $L^* = A^*$

- ▶ $A = \{a, b\}$
- ▶ $L = \{a\}^* \cup \{b\}^* = \{\varepsilon, a, aa, aaa, \dots, b, bb, bbb, \dots\}$
- ▶ Welche Wörter enthält L^* ?
 - ▶ alle Wörter, die man erhält, wenn man
 - ▶ eine beliebige endliche Zahl k
 - ▶ von Wörtern w_1, \dots, w_k aus L
 - ▶ konkateniert zu $w_1 \cdots w_k$
 - ▶ z. B. $aa \cdot \varepsilon \cdot aaaa \cdot b \cdot aaaaa$
 - ▶ z. B. $aa \cdot bbbbb \cdot aaa \cdot b \cdot aaaaa \cdot bbb \cdot aaa$
 - ▶ jedes Wort aus A^* ist die Konkatenation von Blöcken, die nur aus a oder nur aus b bestehen
 - ▶ also ist $L^* = A^*$

- ▶ $A = \{a, b\}$
- ▶ $L = \{a\}^* \cup \{b\}^* = \{\varepsilon, a, aa, aaa, \dots, b, bb, bbb, \dots\}$
- ▶ Welche Wörter enthält L^* ?
 - ▶ alle Wörter, die man erhält, wenn man
 - ▶ eine beliebige endliche Zahl k
 - ▶ von Wörtern w_1, \dots, w_k aus L
 - ▶ konkateniert zu $w_1 \cdots w_k$
 - ▶ z. B. $aa \cdot \varepsilon \cdot aaaa \cdot b \cdot aaaaa$
 - ▶ z. B. $aa \cdot bbbbb \cdot aaa \cdot b \cdot aaaaa \cdot bbb \cdot aaa$
 - ▶ jedes Wort aus A^* ist die Konkatenation von Blöcken, die nur aus a oder nur aus b bestehen
 - ▶ also ist $L^* = A^*$

- ▶ $A = \{a, b\}$
- ▶ $L = \{a\}^* \cup \{b\}^* = \{\varepsilon, a, aa, aaa, \dots, b, bb, bbb, \dots\}$
- ▶ Welche Wörter enthält L^* ?
 - ▶ alle Wörter, die man erhält, wenn man
 - ▶ eine beliebige endliche Zahl k
 - ▶ von Wörtern w_1, \dots, w_k aus L
 - ▶ konkateniert zu $w_1 \cdots w_k$
 - ▶ z. B. $aa \cdot \varepsilon \cdot aaaa \cdot b \cdot aaaaa$
 - ▶ z. B. $aa \cdot bbbbb \cdot aaa \cdot b \cdot aaaaa \cdot bbb \cdot aaa$
 - ▶ jedes Wort aus A^* ist die Konkatenation von Blöcken, die nur aus a oder nur aus b bestehen
 - ▶ also ist $L^* = A^*$

- ▶ $A = \{a, b\}$
- ▶ $L = \{a\}^* \cup \{b\}^* = \{\varepsilon, a, aa, aaa, \dots, b, bb, bbb, \dots\}$
- ▶ Welche Wörter enthält L^* ?
 - ▶ alle Wörter, die man erhält, wenn man
 - ▶ eine beliebige endliche Zahl k
 - ▶ von Wörtern w_1, \dots, w_k aus L
 - ▶ konkateniert zu $w_1 \cdots w_k$
 - ▶ z. B. $aa \cdot \varepsilon \cdot aaaa \cdot b \cdot aaaaa$
 - ▶ z. B. $aa \cdot bbbbb \cdot aaa \cdot b \cdot aaaaa \cdot bbb \cdot aaa$
 - ▶ jedes Wort aus A^* ist die Konkatenation von Blöcken, die nur aus a oder nur aus b bestehen
 - ▶ also ist $L^* = A^*$

- ▶ $A = \{a, b\}$
- ▶ $L = \{a\}^* \cup \{b\}^* = \{\varepsilon, a, aa, aaa, \dots, b, bb, bbb, \dots\}$
- ▶ Welche Wörter enthält L^* ?
 - ▶ alle Wörter, die man erhält, wenn man
 - ▶ eine beliebige endliche Zahl k
 - ▶ von Wörtern w_1, \dots, w_k aus L
 - ▶ konkateniert zu $w_1 \cdots w_k$
 - ▶ z. B. $aa \cdot \varepsilon \cdot aaaa \cdot b \cdot aaaaa$
 - ▶ z. B. $aa \cdot bbbbb \cdot aaa \cdot b \cdot aaaaa \cdot bbb \cdot aaa$
 - ▶ jedes Wort aus A^* ist die Konkatenation von Blöcken, die nur aus a oder nur aus b bestehen
 - ▶ also ist $L^* = A^*$

- ▶ Die Bezeichnung ε -freier Konkatenationsabschluss für L^+ ist irreführend.
 - ▶ Wie steht es um das leere Wort bei L^+ und L^* ?
 - ▶ Klar ist:

$$\varepsilon \in L^0 \subseteq L^*$$

- ▶ Aber: $L = L^1 \subseteq L^+$,
wenn also $\varepsilon \in L$, dann auch $\varepsilon \in L^+$.

- ▶ Beachte

$$\{\}^* = \{\varepsilon\}$$

- ▶ Die Bezeichnung ε -freier Konkatenationsabschluss für L^+ ist irreführend.
 - ▶ Wie steht es um das leere Wort bei L^+ und L^* ?
 - ▶ Klar ist:

$$\varepsilon \in L^0 \subseteq L^*$$

- ▶ Aber: $L = L^1 \subseteq L^+$,
wenn also $\varepsilon \in L$, dann auch $\varepsilon \in L^+$.
- ▶ Beachte

$$\{\}^* = \{\varepsilon\}$$

Das sollten Sie mitnehmen:

- ▶ was **formale Sprachen** sind,
- ▶ wie ihr **Produkt** definiert ist und
- ▶ wie **Konkatenationsabschluss** und ε -freier **Konkatenationsabschluss** definiert sind.

Das sollten Sie üben:

- ▶ Erkennen von Strukturen der Form L^* , L^+ , L_1L_2
- ▶ Lesen von Ausdrücken der Form $(L_1^+L_2)^*$ usw.
- ▶ „Rechnen“ mit formalen Sprachen

Grundbegriffe der Informatik

Einheit 6: Dokumente

Prof. Dr. Tanja Schultz

Karlsruher Institut für Technologie, Fakultät für Informatik

Wintersemester 2011/2012

Dokumente

\LaTeX

XHTML

Eine Grenze unserer bisherigen Vorgehensweise

- ▶ Im alltäglichen Leben gibt es 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**.

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

- ▶ 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**.*

- ▶ 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 unterstützen.
- ▶ **Dokumente:** Texte mit Inhalt, Struktur und Form
 - ▶ z. B. Programme
- ▶ ein **Rat für Ihr weiteres Studium:**
 - ▶ viele Dokumente (PSE, Seminar, Bachelor-Arbeit, ...)
 - ▶ Finden geeigneter Struktur hilft auch eigenem Verständnis
 - ▶ *deswegen: früh damit beginnen, etwas aufzuschreiben*
(weil man dann über Struktur nachdenken muss)

Wozu Inhalt, Struktur und Form?

- ▶ Inhalt üblicherweise für Autoren und Leser im Vordergrund
 - ▶ (Ausnahmen?)
- ▶ Struktur und Form
 - ▶ sollen den Leser beim Verstehen des Inhalts unterstützen.
- ▶ **Dokumente:** Texte mit Inhalt, Struktur und Form
 - ▶ z. B. Programme
- ▶ ein Rat für Ihr weiteres Studium:
 - ▶ viele Dokumente (PSE, Seminar, Bachelor-Arbeit, ...)
 - ▶ Finden geeigneter Struktur hilft auch eigenem Verständnis
 - ▶ *deswegen: früh damit beginnen, etwas aufzuschreiben*
(weil man dann über Struktur nachdenken muss)

Wozu Inhalt, Struktur und Form?

- ▶ Inhalt üblicherweise für Autoren und Leser im Vordergrund
 - ▶ (Ausnahmen?)
- ▶ Struktur und Form
 - ▶ sollen den Leser beim Verstehen des Inhalts unterstützen.
- ▶ **Dokumente:** Texte mit Inhalt, Struktur und Form
 - ▶ z. B. Programme
- ▶ ein **Rat für Ihr weiteres Studium:**
 - ▶ viele Dokumente (PSE, Seminar, Bachelor-Arbeit, ...)
 - ▶ Finden geeigneter Struktur hilft auch eigenem Verständnis
 - ▶ deswegen: *früh damit beginnen, etwas aufzuschreiben*
(weil man dann über Struktur nachdenken muss)

- ▶ 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
 - ▶ Hypertext Markup Language (HTML; dt. Hypertext-Auszeichnungssprache) ist eine textbasierte Auszeichnungssprache zur Strukturierung von Inhalten wie Texten, Bildern und Hyperlinks in Dokumenten.
 - ▶ XHTML eXtensible HTML – erweiterbare HTML

Dokumente

L^AT_EX

XHTML

Eine Grenze unserer bisherigen Vorgehensweise

- ▶ basiert auf T_EX 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

$$\left[2 - \sum_{i=0}^k i 2^{-i} = (k+2) 2^{-k} \right]$$
 - ▶ wird

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

- ▶ 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
`\section{Struktur von Dokumenten}`
- ▶ woraus LaTeX die Zeile

7.2 STRUKTUR VON DOKUMENTEN

auf Seite 50 des Skriptes gemacht hat

- ▶ also
 - ▶ automatisch die passende Abschnittsnummer eingefügt
 - ▶ alles wurde in Großbuchstaben gesetzt
- ▶ Beachte: z. B. Schriftauswahl ist *nicht* in der Eingabe mit vermerkt.
- ▶ Diese Festlegung findet sich an anderer Stelle, und zwar an *einer* Stelle, an der das typografische Aussehen *aller* Abschnittsüberschriften (einheitlich) festgelegt ist.

```
\documentclass[11pt]{report}
% so schreibt man Kommentare
% dieser Teil heit Prambel des Dokumentes
% Untersttzung fr Deutsch,
% z.B. richtige automatische Trennung
  \usepackage[german]{babel}
  % Angabe des Zeichensatzes, in dem der Text ist
  \usepackage[latin1]{inputenc}
  % fr das Einbinden von Grafiken
  \usepackage{graphicx}
\begin{document}
  % und hier kommt der eigentliche Text .....
\end{document}
```

- ▶ Eine Liste einfacher Punkte sieht in L^AT_EX so aus:

Eingabe	Ausgabe
<code>\begin{itemize}</code>	
<code>\item Inhalt</code>	• Inhalt
<code>\item Struktur</code>	• Struktur
<code>\item Form</code>	• Form
<code>\end{itemize}</code>	

- ▶ der dicke 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. „–“ statt „•“ möchte, dann muss man an *einer* Stelle (in der Präambel) die Definition `\item` ändern.

- ▶ 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
$$L_{\text{itemize}} = \{\texttt{\backslash begin\{itemize\}}\} \left(\{\texttt{\backslash item}\} L_{\text{item}} \right)^+ \{\texttt{\backslash end\{itemize\}}\}$$
- ▶ Problem: in Aufzählungspunkt wieder eine Liste erlaubt; also

$$\begin{aligned} L_{\text{item}} &= \dots L_{\text{itemize}} \dots \\ L_{\text{itemize}} &= \dots L_{\text{item}} \dots \end{aligned}$$

Dokumente

L^AT_EX

XHTML

Eine Grenze unserer bisherigen Vorgehensweise

- ▶ 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*.

```
<!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;>
```



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

- ▶ **table**, **thead**, **tr**: Namen für formale Sprachen
- ▶ Bedeutung von **+** ε -freier Konkatenationsabschluss
- ▶ Bedeutung von **,** Produkt formaler Sprachen
- ▶ Bedeutung von **|** Vereinigung
- ▶ Fragezeichen ist neu, aber ganz einfach:

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

- ▶ „optionales“ Auftreten eines Wortes aus L

- ▶ Mitteilung:

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

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

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

$$L_{\text{table}} = \{\text{<table>}\} \cdot L_{\text{caption}}^? \cdot L_{\text{thead}}^? \cdot L_{\text{tfoot}}^? \cdot L_{\text{tbody}}^+ \cdot \{\text{</table>}\}$$

- ▶ 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 Tabellenrumpfen.

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

Dokumente

L^AT_EX

XHTML

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 ...

- ▶ 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 ...

- ▶ anderes typisches Beispiel: korrekte Klammerungen (wie bei arithmetischen Ausdrücken)
- ▶ Bei einer syntaktisch korrekten Klammerung gibt es zu jeder *Klammer auf* „weiter hinten“ die „zugehörige“ *Klammer zu*.
- ▶ insbesondere legal:
 - ▶ mehrere korrekte Klammerungen hintereinander
 - ▶ zusätzliche Klammern um korrekte Klammerung außen herum
- ▶ schön wäre z. B. Beziehung von L_{Klammer} mit L_{Klammer}^* und mit $\{ () \cdot L_{\text{Klammer}} \cdot \} \}$
- ▶ aber wie? Gleichung?
 - ▶ lösbar?
 - ▶ wenn ja: eindeutig?
 - ▶ ein Thema des nächsten Kapitels ...

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