

Sprachen & Grammatiken: Einführung

Gemeinsame Grundkonzepte für formale und natürliche Sprachen

Das **Alphabet**: die Basisbausteine aus denen **Wörter** gebildet werden,

z.B. $A, B, C, \dots, a, b, c, \dots, 0, 1, 2, \dots$

Die Menge der **Wörter**: die Basisbausteine aus denen **Sätze** gebildet werden

z.B. *sprache, Sprache, SPRACHE, sprACHE, ...*

Grammatische **Regeln**:

die Bedingungen / Konstruktionsbeschreibungen dafür, wie die Basisbausteine zu **komplexen sprachlichen Ausdrücken** kombiniert werden können, d.h. welche Zeichenketten als „wohlgeformt“ gelten.

- In natürlichen Sprachen spielen alle drei Ebenen, Alphabete, Wörter, Sätze eine Rolle.
- Für formale Sprachen werden überwiegend nur zwei Ebenen betrachtet.
- Formale Sprachen und formale Grammatiken werden auch für die Spezifikation natürlicher Sprachen verwendet.

Alphabet

Definition 1.1

Ein **Alphabet** ist eine (total geordnete) endliche Menge von unterschiedlichen Zeichen (oder Symbolen).

Anmerkungen

1. Im Weiteren werden Alphabete (meist) durch grosse griechische Buchstaben bezeichnet, im Standardfall durch Σ (bzw. $\Sigma_1, \Sigma_2, \dots$)
2. Die (lexikographische / alphabetische) Ordnung der Elemente eines Alphabets ist nützlich, um bei Verfahren/Algorithmen und Beweisen (zu Verfahren) systematisch alle relevanten Zeichenketten behandeln zu können.

Beispiele

$$\Sigma_1 = \{ a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z \}$$

$$\Sigma_2 = \{ a, b, c, \dots, x, y, z, A, B, C, \dots, X, Y, Z \}$$

$$\Sigma_3 = \{ 0, 1, 2, 3, 3, 5, 6, 7, 8, 9 \}$$

$$\Sigma_4 = \{ 0, 1 \}$$

$$\Sigma_5 = \{ \alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \iota, \kappa, \lambda, \mu, \nu, \xi, \omicron, \pi, \rho, \sigma, \tau, \upsilon, \phi, \chi, \psi, \omega \}$$

Wortmengen

Definition 1.2

1. Für ein Alphabet Σ ist (Σ^*, \circ) das freie Monoid mit der **Konkatenation** (\circ) oder Hintereinanderschreibung der einzelnen Zeichen als Monoid-Operation, und dem leeren Wort ε (Epsilon) als neutralem Element.
 Σ^* bezeichnet die Menge aller **endlichen Wörter** über dem Alphabet Σ .
 $\Sigma^+ = \Sigma^* \setminus \{ \varepsilon \}$ bezeichnet die Menge aller **endlichen Wörter** über dem Alphabet Σ , die ungleich dem leeren Wort sind.
2. Für $w \in \Sigma^*$ schreiben wir w^k anstelle von $\underbrace{w \circ w \dots \circ w}_k$. Es sei $w^0 := \varepsilon$.
3. Jede Menge $L \subseteq \Sigma^*$ heißt **formale Sprache**.
4. Die Anzahl der Symbole, die ein Wort w enthält, wird als die Länge des Wortes bezeichnet, und durch $|w|$ dargestellt

Anmerkungen

- $|\varepsilon| = 0$
- Statt $w_1 \circ w_2$ wird häufig auch $w_1 w_2$ geschrieben.
- Wörter werden auch als endliche Sequenzen von Symbolen angesehen.

Wörter & Wortmengen – Beispiele

$$\Sigma_3 = \{ 0, 1, 2, 3, 3, 5, 6, 7, 8, 9 \}$$

$$101, 91, 7777 \in \Sigma_3^*$$

$$\Sigma_4 = \{ 0, 1 \}$$

$$101 \in \Sigma_4^*, \quad 91, 7777 \notin \Sigma_4^*$$

$$\Sigma_1 = \{ a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z \}$$

$$\Sigma_2 = \{ a, b, c, \dots, x, y, z, A, B, C, \dots, X, Y, Z \}$$

$$\text{wortmengen} \in \Sigma_1^*, \quad \text{wortmengen, Wortmengen} \in \Sigma_2^*$$

Alphabete und Wortmengen (2)

- Häufig wird das Alphabet auch als **Vokabular** bezeichnet, und dann mit dem Symbol V benannt. In diesem Fall wird dann V^* als **Satzmenge** bezeichnet.
- Die leere Menge \emptyset und die Menge $\{\varepsilon\}$ sind (formale) Sprachen. (Sie sind verschieden!)
- Jede formale Sprache über dem Alphabet Σ ist Teilmenge (Teilsprache) von Σ^* . Die Theorie der Formalen Sprachen und Formalen Grammatiken befasst sich damit, „interessante Teilsprachen“ bzw. Klassen von Teilsprachen zu beschreiben. Eine Formale Grammatik ist in dieser Hinsicht ein System, das eine Sprache charakterisiert. Insbesondere werden auch nicht-endliche Sprachen, also nicht-endliche Mengen von Wörtern, bzw. Sätzen, untersucht.
[Nicht-endliche Sprachen korrespondieren zur Produktivität natürlicher Sprachen.]

Konkatenation

Wichtige Eigenschaften der Konkatenation

Die Konkatenation $\circ : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$ ist

1. assoziativ: $(a \circ b) \circ c = a \circ (b \circ c)$
2. nicht kommutativ, denn $a \circ b$ ist verschieden zu $b \circ a$

Für das leere Wort ε gilt:

$$w = w \circ \varepsilon = \varepsilon \circ w \text{ für beliebige Wörter } w$$

D. h. ε ist das neutrale Element der Konkatenation.

Teilwörter, Präfixe und Suffixe

Wenn $u \neq \varepsilon$, $w \neq \varepsilon$, w_{pr} und w_{suf} Zeichenketten über Σ sind, für die

$$u = w_{pr} \circ w \circ w_{suf}$$

gilt, so heißt w **Teil(-wort)** von u , w_{pr} heißt **Präfix** (Wortanfang) und w_{suf} **Suffix** (Wortende). Ist $w \neq u$, so wird w als echtes Teilwort von u bezeichnet.

Anmerkungen

- Das leere Wort ist Präfix und Suffix für jedes Wort.
- Als echte Präfixe / Suffixe werden solche Präfixe / Suffixe bezeichnet, die nicht mit dem Gesamtwort identisch sind.
- Echte Präfixe / Suffixe sind Spezialfälle von Teilwörtern

| | | |
|-----------|----------------|----------------|
| Beispiel: | <i>Präfixe</i> | <i>Suffixe</i> |
| | T | eilwort |
| | Te | ilwort |
| | Tei | lwort |
| | Teil | wort |
| | Teilw | ort |
| | Teilwo | rt |
| | Teilwor | t |

Konkatenation von Sprachen

Definition 1.3

Seien $L_1 \subseteq \Sigma_1^*$ und $L_2 \subseteq \Sigma_2^*$ zwei Sprachen (über eventuell verschiedenen Alphabeten).

Die Konkatenation (das Produkt) dieser Sprachen ist definiert durch:

$$L_1 \circ L_2 = \{ w_1 \circ w_2 \mid w_1 \in L_1 \text{ und } w_2 \in L_2 \}$$

[häufig wird auch $L_1 L_2$ geschrieben.]

Beispiel

$$L_1 = \{A, B, \dots, X, Y, Z\} \quad L_2 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$L_1 \circ L_2 = \{ w_1 \circ w_2 \mid w_1 \in L_1 \text{ und } w_2 \in L_2 \} = \{ A0, A1, \dots, L7, \dots \}$$

Satz 1.4

Die Konkatenation über Sprachen besitzt sowohl ein Eins- als auch ein Null-Element:

$$(i) \quad \{ \varepsilon \} \circ L = L \circ \{ \varepsilon \} = L$$

$$(ii) \quad \emptyset \circ L = L \circ \emptyset = \emptyset$$

Mengentheoretische Operationen über Sprachen

Seien $L_1 \subseteq \Sigma_1^*$ und $L_2 \subseteq \Sigma_2^*$ zwei Sprachen (über eventuell verschiedenen Alphabeten).

- $L_1 \cup L_2$ und $L_1 \cap L_2$ sind Sprachen [über $(\Sigma_1 \cup \Sigma_2)^*$ bzw. über $(\Sigma_1 \cap \Sigma_2)^*$], sie werden als Vereinigung bzw. Durchschnitt von L_1 und L_2 bezeichnet.
- $\Sigma_1^* \setminus L_1$ heisst das Komplement von L_1 .

Anmerkung

- In der Theorie der Formalen Sprachen, Automaten & Maschinen wird u.a. untersucht, welche Eigenschaften von formalen Sprachen unter derartigen Operationen erhalten bleiben, d.h. ob für eine Eigenschaft gilt:
Wenn L_1 und L_2 die Eigenschaft p besitzen, besitzen dann auch die Vereinigung, der Durchschnitt und das Komplement die Eigenschaft p ?

Weitere Operationen über Sprachen

Definition 1.5

Sei L eine Sprache über Σ .

Für jedes $n \in \mathbb{N}$ wird festgelegt:

$$L^1 = L, \quad L^{n+1} = L \circ L^n$$

Entsprechend gilt: $L^0 = \{ \varepsilon \}$

$L^* = \bigcup_{i=0}^{\infty} L^i$ heisst der Abschluss (oder die Hülle) der Sprache L .

Zahlensysteme als Beispiele

Dezimaldarstellung natürlicher Zahlen

Alphabet ist die Ziffernmenge: $B = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Wörter aus B^* sind z.B.:

1, 13, 250, 01,, aber auch ε

Variante 1:

Wir schließen ε als zulässiges Wort der Sprache \mathcal{N} , der Dezimaldarstellungen von natürlichen Zahlen (einschliesslich Null), d.h. Zahlen aus \mathbb{N}_0 , aus.

$$\mathcal{N}_1 := B^+$$

Variante 2:

Wir beabsichtigen die Ziffer ‚0‘ nicht als echtes Präfix einer Dezimaldarstellung zuzulassen, d.h. ‚01‘ nicht als „wohlgeformte“ Dezimaldarstellung der natürlichen Zahl ‚1‘ aufzufassen.

$$\mathcal{N}_2 := \{0\} \cup (B \setminus \{0\} \circ B^*)$$

Fragestellungen der Theorie formaler Sprachen

Wie sind Sprachen spezifiziert?

extensionale Darstellung: Auflistung der Elemente

intensionale Darstellung: mengentheoretische Charakterisierung über Eigenschaften

Charakterisierung über

- Mechanismen, die Sprachen erzeugen: Grammatiken
- Mechanismen, die entscheiden, ob Sätze zu einer Sprache gehören: Automaten / Maschinen.

Wie können sprachliche Strukturen berechnet werden?

Syntaxanalyse

Kontextfreie Grammatiken

Definition 1.6

Eine **kontextfreie Grammatik** G ist ein 4-Tupel (Σ, N, P, S) , für das gilt:

- Σ ist ein Alphabet, genannt das Alphabet der **Terminalsymbole**
- N ist ein Alphabet (von **Nichtterminalsymbolen**), das disjunkt zu Σ ist
- P ist eine endliche Menge von Produktionsregeln (auch als **Regeln** bezeichnet), wobei jede Regel ein Paar (A, w) ist, mit $A \in N$ und $w \in (\Sigma \cup N)^*$
- $S \in N$ heißt **Startsymbol**

Anmerkung

- Kontextfreie Grammatik wird auch durch kfG oder CFG abgekürzt.
- Nichtterminale werden auch als Variable bezeichnet.
- Regeln werden im Weiteren in der Form $A \rightarrow w$ geschrieben.
- A wird als linke und w als rechte Seite der Regel bezeichnet.
- Regeln, die die gleiche linke Seite haben, d.h. die Ableitungen vom gleichen nichtterminalen Symbol betreffen, werden häufig „zusammengefasst“ (siehe Folie 1-19)
- Auch $A \rightarrow \varepsilon$ ist eine zulässige Regel (für kontextfreie Grammatiken)

Regelanwendung, Ableitung

Definition 1.7

Seien u, v, w Zeichenketten über $(\Sigma \cup N)$, und $A \rightarrow w$ eine Regel (einer Grammatik).

Durch die **Anwendung der Regel** kann aus dem Wort uAv das Wort uwv (direkt) abgeleitet werden.

Man sagt auch: Die Regel $A \rightarrow w$ führt vom Wort uAv zum Wort uwv , bzw. das Nichtterminal / die Variable A wird durch die Regel zu w **expandiert**.

- Die Regelanwendung wird auch als Ableitung (in einem Schritt) bezeichnet, und als $uAv \Rightarrow uwv$ geschrieben.

Wenn $u = v$ oder wenn eine Folge $u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \dots \Rightarrow u_k \Rightarrow v$ existiert (mit $k \geq 0$), so ist v aus u (in gegebenenfalls mehreren Schritten) ableitbar. Dieses wird durch $u \xRightarrow{*} v$ notiert. Die Sequenz $u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \dots \Rightarrow u_k \Rightarrow v$ wird als **Ableitung** (derivation) bezeichnet.

- Derartige Ableitungen sind stets von endlicher Länge (endliche Anzahl von Schritten); die Ableitungssequenz kann aber beliebige Länge haben.
- Zeichenketten w über $(\Sigma \cup N)$, für die eine Ableitung $S \xRightarrow{*} w$, existiert, werden als **Satzformen** bezeichnet.

Zum Selbststudium

In der Theoretischen Informatik, Logik und Mathematik werden „gerichtete Beziehungen“, z.B. asymmetrische oder antisymmetrische Relationen, häufig durch Pfeile dargestellt. Um innerhalb eines Gebietes verschiedene derartige Relationen unterscheiden zu können, werden unterschiedliche Pfeile verwendet. Hier etwa, um Regel „ \rightarrow “ im Gegensatz zu Regelanwendung/Ableitung „ \Rightarrow “

Da das Darstellungsinventar von Pfeilen recht beschränkt ist, kann ein Pfeiltyp in verschiedenen Verwendungen auftreten. (VORSICHT!)

In der Vorlesung FGI-1 werden für die Logik ebenfalls Pfeile verwendet, die jedoch andere Bedeutungen und Anwendungsbereiche haben.

Von einer Grammatik erzeugte Sprache

Definition 1.8

Sei $G = (\Sigma, N, P, S)$ eine kontextfreie Grammatik, so ist

$$L(G) = \{ w \in \Sigma^* \mid S \xRightarrow{*} w \}$$

die von G erzeugte Sprache.

Anmerkung

- $L(G)$ umfasst also genau die Wörter über dem terminalen Alphabet Σ , für die eine Ableitung (endlicher Länge) besteht, die beim Startsymbol S beginnt.

Beispiel

$$G_1 = (\{a, b\}, \{S\}, P, S) \text{ mit}$$

$$P = \{ S \rightarrow aSb, S \rightarrow ab \}$$

Einige Ableitungen:

$$S \Rightarrow ab$$

$$S \Rightarrow aSb \Rightarrow aabb$$

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$$

$$L(G_1) = \{ w \in \Sigma^* \mid w = a^n b^n, n \geq 1 \} = \{ a^n b^n \mid n \geq 1 \}$$

Beweis als Nachbereitungsaufgabe.

Grammatik für $\{ a^n b^n \mid n \geq 0 \}$

$G_2 = (\{a, b\}, \{S\}, P, S)$ mit

$P = \{ S \rightarrow aSb, S \rightarrow \varepsilon \}$

$S \Rightarrow \varepsilon$

$S \Rightarrow aSb \Rightarrow ab$

$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$

$L(G_2) = \{ a^n b^n \mid n \geq 0 \}$, also $\varepsilon \in L(G_2)$

Ableitungsbäume / Strukturbäume (Parse trees)

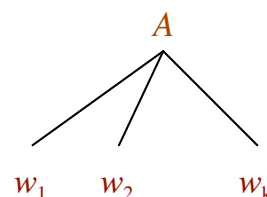
Sei $A \rightarrow w$ eine Regel, mit $|w| = k$,

wobei $w = w_1 w_2 \dots w_k$ die Darstellung durch Symbole des Alphabets ist.

Dann existiert ein zu $A \rightarrow w_1 w_2 \dots w_k$ korrespondierender Baum

des Verzweigungsgrades k mit Tiefe 2,

der Wurzel A und den Blättern w_1, w_2, \dots, w_k .



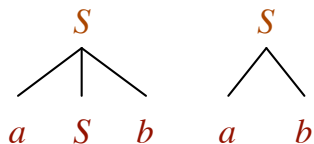
Sei $S \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \dots \Rightarrow u_k \Rightarrow w$ die Ableitung

eines Wortes $w \in L(G)$, so kann ein Ableitungsbaum (**Strukturbaum**) zu dieser

Ableitung gebildet werden, indem die zu den verwendeten Regeln korrespondierenden Bäume „konkateniert“ werden.

Beispiel: $G_1 = (\{a, b\}, \{S\}, P, S)$

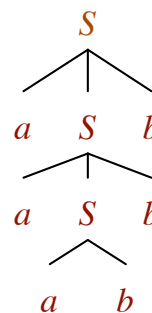
mit $P = \{ S \rightarrow aSb, S \rightarrow ab \}$



$S \Rightarrow aSb$

$\Rightarrow aaSbb$

$\Rightarrow aaabbb$



Beispiel (Grammatik G_3): Arithmetische Ausdrücke

$$G_3 = (\Sigma, N, P, \langle \text{EXPR} \rangle)$$

$$\Sigma = \{ a, +, \times, (,) \}$$

$$N = \{ \langle \text{EXPR} \rangle, \langle \text{TERM} \rangle, \langle \text{FACTOR} \rangle \}$$

$$P = \{ \langle \text{EXPR} \rangle \rightarrow \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle, \langle \text{EXPR} \rangle \rightarrow \langle \text{TERM} \rangle, \\ \langle \text{TERM} \rangle \rightarrow \langle \text{TERM} \rangle \times \langle \text{FACTOR} \rangle, \langle \text{TERM} \rangle \rightarrow \langle \text{FACTOR} \rangle, \\ \langle \text{FACTOR} \rangle \rightarrow (\langle \text{EXPR} \rangle), \langle \text{FACTOR} \rangle \rightarrow a \}$$

$$a + a \times a \in L(G_3)$$

$$(a + a) \times a \in L(G_3)$$

aber die Ableitungen führen zu verschiedenen Strukturbäumen.

- Die Grammatik spiegelt Präferenz „Punktrechnung vor Strichrechnung“ wider. [Überzeugen Sie sich hiervon bei der Nachbereitung durch Konstruktion der Bäume.]

Die Regeln in zusammengefasster Form (Backus-Naur-Form):

$$P = \{ \langle \text{EXPR} \rangle \rightarrow \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle \mid \langle \text{TERM} \rangle, \\ \langle \text{TERM} \rangle \rightarrow \langle \text{TERM} \rangle \times \langle \text{FACTOR} \rangle, \mid \langle \text{FACTOR} \rangle, \\ \langle \text{FACTOR} \rangle \rightarrow (\langle \text{EXPR} \rangle) \mid a \}$$

Beispiel (Grammatik G_4): Arithmetische Ausdrücke II

$$G_4 = (\Sigma, N, P, \langle \text{EXPR} \rangle)$$

$$\Sigma = \{ a, +, \times, (,) \} \quad N = \{ \langle \text{EXPR} \rangle \}$$

$$P = \{ \langle \text{EXPR} \rangle \rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle \mid \langle \text{EXPR} \rangle \times \langle \text{EXPR} \rangle \mid (\langle \text{EXPR} \rangle) \mid \rightarrow a \}$$

$$L(G_4) = L(G_3)$$

Nachprüfen!!

Verschiedene **Ableitungen** für $a + a + a \in L(G_4)$

| | |
|---|--|
| $\langle \text{EXPR} \rangle \Rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle$ | $\langle \text{EXPR} \rangle \Rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle$ |
| $\Rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle$ | $\Rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle$ |
| $\Rightarrow a + \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle$ | $\Rightarrow a + \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle$ |
| $\Rightarrow a + a + \langle \text{EXPR} \rangle$ | $\Rightarrow a + \langle \text{EXPR} \rangle + a$ |
| $\Rightarrow a + a + a$ | $\Rightarrow a + a + a$ |
| $\langle \text{EXPR} \rangle \Rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle$ | ➤ Es gibt – im Allgemeinen – keine Festlegung, welches Nichtterminal durch die Regelanwendung expandiert wird. |
| $\Rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle$ | |
| $\Rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle + a$ | |
| $\Rightarrow \langle \text{EXPR} \rangle + a + a$ | |
| $\Rightarrow a + a + a$ | |

Satzformen betreffen (Zwischen-)resultate des Ableitungsprozesses.

Linksableitung – Rechtsableitung

Definition 1.9

Wird in einer Ableitung (Ableitungssequenz) stets die am weitesten links (rechts) auftretende Variable expandiert, so wird die Ableitung als **Linksableitung** / Rechtsableitung (leftmost / rightmost derivation) bezeichnet.

Wir verwenden die Symbole \Rightarrow_{lm} bzw. \Rightarrow_{rm} für Ableitungsschritte und \Rightarrow_{lm}^* bzw. \Rightarrow_{rm}^* für Ableitungssequenzen, die bzgl. *leftmost* oder *rightmost* festgelegt sind.

Linksableitung

$$\begin{aligned} \langle \text{EXPR} \rangle &\Rightarrow_{lm} \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle \\ &\Rightarrow_{lm} \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle \\ &\Rightarrow_{lm} a + \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle \\ &\Rightarrow_{lm} a + a + \langle \text{EXPR} \rangle \\ &\Rightarrow_{lm} a + a + a \end{aligned}$$

Rechtsableitung

$$\begin{aligned} \langle \text{EXPR} \rangle &\Rightarrow_{rm} \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle \\ &\Rightarrow_{rm} \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle \\ &\Rightarrow_{rm} \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle + a \\ &\Rightarrow_{rm} \langle \text{EXPR} \rangle + a + a \\ &\Rightarrow_{rm} a + a + a \end{aligned}$$

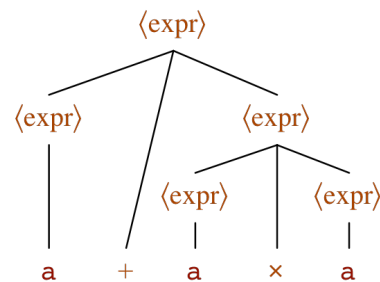
Satz (ohne Beweis): Zu jeder Ableitung existiert eine äquivalente Linksableitung und eine äquivalente Rechtsableitung. D.h.: Für eine Zeichenkette v gilt $u \Rightarrow^* v$ genau dann, wenn $u \Rightarrow_{lm}^* v$ und genau dann, wenn $u \Rightarrow_{rm}^* v$.

Arithmetische Ausdrücke in $L(G_4)$: Strukturbäume

Für $a + a \times a \in L(G_4)$ gibt es Ableitungen mit unterschiedlichen Ableitungsbäumen. Die unterschiedlichen Strukturbäume entsprechen unterschiedlichen Bedeutungen.

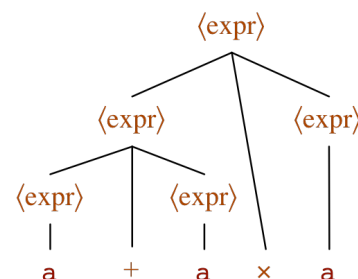
$$\begin{aligned} \langle \text{EXPR} \rangle &\Rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle \\ &\Rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle \times \langle \text{EXPR} \rangle \\ &\Rightarrow a + \langle \text{EXPR} \rangle \times \langle \text{EXPR} \rangle \\ &\Rightarrow a + a \times \langle \text{EXPR} \rangle \\ &\Rightarrow a + a \times a \end{aligned}$$

ist keine Linksableitung



$$\begin{aligned} \langle \text{EXPR} \rangle &\Rightarrow \langle \text{EXPR} \rangle \times \langle \text{EXPR} \rangle \\ &\Rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle \times \langle \text{EXPR} \rangle \\ &\Rightarrow a + \langle \text{EXPR} \rangle \times \langle \text{EXPR} \rangle \\ &\Rightarrow a + a \times \langle \text{EXPR} \rangle \\ &\Rightarrow a + a \times a \end{aligned}$$

ist eine Linksableitung



Mehrdeutigkeit (Ambiguität)

Definition 1.10

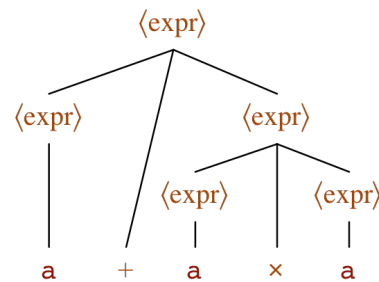
Eine Zeichenkette w ist **mehrdeutig**, bzw. wird durch eine kfG G mehrdeutig abgeleitet, falls w zwei (oder mehr) verschiedene Linksableitungen besitzt.

Eine Grammatik G ist mehrdeutig, falls es Wörter $w \in L(G)$ gibt, die mehrdeutig sind.

$a + a \times a \in L(G_4)$

$\langle \text{EXPR} \rangle \Rightarrow_{lm} \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle$
 $\Rightarrow_{lm} a + \langle \text{EXPR} \rangle$
 $\Rightarrow_{lm} a + \langle \text{EXPR} \rangle \times \langle \text{EXPR} \rangle$
 $\Rightarrow_{lm} a + a \times \langle \text{EXPR} \rangle$
 $\Rightarrow_{lm} a + a \times a$

ist eine Linksableitung



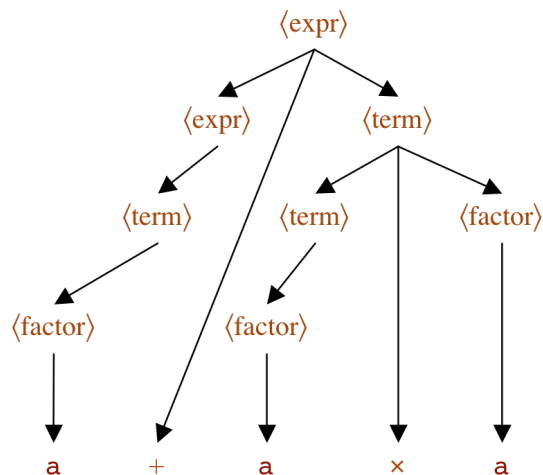
→ Die Grammatik G_4 ist mehrdeutig.

Zum Selbststudium: Arithmetische Ausdrücke in $L(G_3)$

$G_3 = (\Sigma, N, P, \langle \text{EXPR} \rangle)$ $\Sigma = \{ a, +, \times, (,) \}$ $N = \{ \langle \text{EXPR} \rangle, \langle \text{TERM} \rangle, \langle \text{FACTOR} \rangle \}$
 $P = \{ \langle \text{EXPR} \rangle \rightarrow \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle \mid \langle \text{TERM} \rangle,$
 $\langle \text{TERM} \rangle \rightarrow \langle \text{TERM} \rangle \times \langle \text{FACTOR} \rangle \mid \langle \text{FACTOR} \rangle,$
 $\langle \text{FACTOR} \rangle \rightarrow (\langle \text{EXPR} \rangle) \mid a \}$

$\langle \text{EXPR} \rangle \Rightarrow_{lm} \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle$
 $\Rightarrow_{lm} \langle \text{TERM} \rangle + \langle \text{TERM} \rangle$
 $\Rightarrow_{lm} \langle \text{FACTOR} \rangle + \langle \text{TERM} \rangle$
 $\Rightarrow_{lm} a + \langle \text{TERM} \rangle$
 $\Rightarrow_{lm} a + \langle \text{TERM} \rangle \times \langle \text{FACTOR} \rangle$
 $\Rightarrow_{lm} a + \langle \text{FACTOR} \rangle \times \langle \text{FACTOR} \rangle$
 $\Rightarrow_{lm} a + a \times \langle \text{FACTOR} \rangle$
 $\Rightarrow_{lm} a + a \times a$

ist eine Linksableitung



Zum Selbststudium: Arithmetische Ausdrücke in $L(G_3)$ Forts.

- Führen Sie eine Linksableitung der Zeichenkette $(a + a) \times a$ durch und konstruieren Sie den korrespondierenden Strukturbaum.
- Machen Sie sich klar, inwiefern die beiden Strukturbäume zu $a + a \times a$ bzw. zu $(a + a) \times a$ zu unterschiedlichen Auswertungen, d.h. Berechnungen der Werte der arithmetischen Ausdrücke führen.

Die Grammatik G_3 ist so entworfen, dass Zeichenketten eindeutig sind und somit eine eindeutige Bedeutung haben.

Parsing

Das Parsingproblem:

Gegeben eine kontextfreie Grammatik G und eine Zeichenkette w .

- Jede Ableitung $S \xRightarrow{*} w$ bestimmt einen Strukturbaum, d.h. eine syntaktische Struktur, zu w .

Die Parsingaufgabe: Bestimme die syntaktische(n) Struktur(en) von w bzgl. G .

Anmerkungen

- Wenn G nicht mehrdeutig ist, dann hat jedes Wort $w \in L(G)$ genau einen korrespondierenden Strukturbaum.
- Der Prozess des Parsings weist nur Wörtern aus $L(G)$ syntaktische Strukturen zu, d.h. für Zeichenketten $w \notin L(G)$ sollte der Parser die Nichtzugehörigkeit zu $L(G)$ ausweisen.
- Parsing ist – in gewisser Weise – eine Umkehrung der Generierung von Zeichenketten bei gleichzeitiger Zuweisung der syntaktischen Struktur.

Wir werden in einem späteren Abschnitt detaillierter aufs Parsing eingehen.

Parsing Beispiel: Arithmetische Ausdrücke in $L(G_3)$

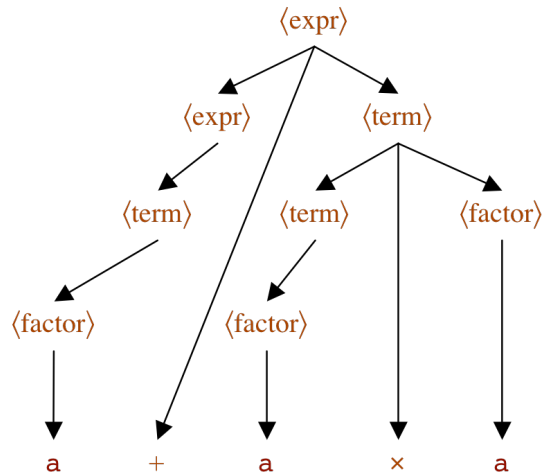
$$G_3 = (\Sigma, N, P, \langle \text{EXPR} \rangle) \quad \Sigma = \{ a, +, \times, (,) \} \quad N = \{ \langle \text{EXPR} \rangle, \langle \text{TERM} \rangle, \langle \text{FACTOR} \rangle \}$$

$$P = \{ \langle \text{EXPR} \rangle \rightarrow \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle \mid \langle \text{TERM} \rangle, \\ \langle \text{TERM} \rangle \rightarrow \langle \text{TERM} \rangle \times \langle \text{FACTOR} \rangle \mid \langle \text{FACTOR} \rangle, \\ \langle \text{FACTOR} \rangle \rightarrow (\langle \text{EXPR} \rangle) \mid a \}$$

$a + a \times a$

- $\Leftarrow \langle \text{FACTOR} \rangle + a \times a$
- $\Leftarrow \langle \text{TERM} \rangle + a \times a$
- $\Leftarrow \langle \text{EXPR} \rangle + a \times a$
- $\Leftarrow \langle \text{EXPR} \rangle + \langle \text{FACTOR} \rangle \times a$
- $\Leftarrow \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle \times a$
- $\Leftarrow \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle \times \langle \text{FACTOR} \rangle$
- $\Leftarrow \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle$
- $\Leftarrow \langle \text{EXPR} \rangle$

[erfolgreicher Parse,
aber erst nach Backtracking]



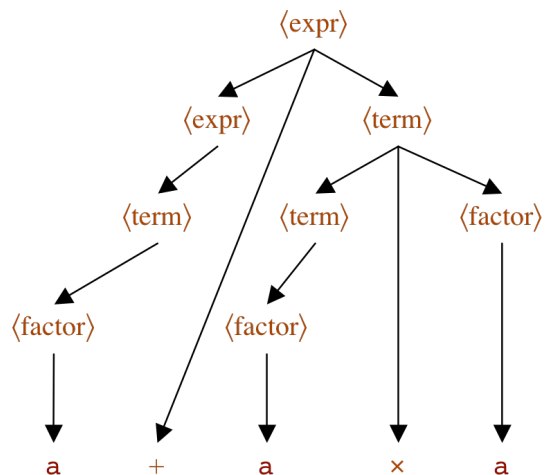
Parsing Beispiel: Arithmetische Ausdrücke in $L(G_3)$ – Forts.

$$G_3 = (\Sigma, N, P, \langle \text{EXPR} \rangle) \quad \Sigma = \{ a, +, \times, (,) \} \quad N = \{ \langle \text{EXPR} \rangle, \langle \text{TERM} \rangle, \langle \text{FACTOR} \rangle \}$$

$$P = \{ \langle \text{EXPR} \rangle \rightarrow \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle \mid \langle \text{TERM} \rangle, \\ \langle \text{TERM} \rangle \rightarrow \langle \text{TERM} \rangle \times \langle \text{FACTOR} \rangle \mid \langle \text{FACTOR} \rangle, \\ \langle \text{FACTOR} \rangle \rightarrow (\langle \text{EXPR} \rangle) \mid a \}$$

$a + a \times a$

- $\Leftarrow \langle \text{FACTOR} \rangle + a \times a$
- $\Leftarrow \langle \text{TERM} \rangle + a \times a$
- $\Leftarrow \langle \text{EXPR} \rangle + a \times a$
- $\Leftarrow \langle \text{EXPR} \rangle + \langle \text{FACTOR} \rangle \times a$
- $\Leftarrow \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle \times a$
- $\Leftarrow \langle \text{EXPR} \rangle \times a$
- $\Leftarrow \langle \text{EXPR} \rangle \times \langle \text{FACTOR} \rangle$
- $\Leftarrow \langle \text{EXPR} \rangle \times \langle \text{TERM} \rangle$
- $\Leftarrow \langle \text{EXPR} \rangle \times \langle \text{EXPR} \rangle$
- $\Leftarrow \#$ [Backtracking notwendig!]



Kontextfreie Grammatiken – Regelformen

Die Regeln $A \rightarrow w$ einer Kontextfreien Grammatik haben die Form

- $A \in N$ und $w \in (\Sigma \cup N)^*$, d.h.
- linke Seite ein Nichtterminal, rechte Seite eine beliebige Kette über $(\Sigma \cup N)^*$

In der Theorie der formalen Sprachen wird u.a. untersucht

- inwieweit unterschiedliche Bedingungen an die Regelform, unterschiedliche Sprachklassen definierten
- **reguläre Sprachen** können über kfG mit spezifischer Regelform spezifiziert werden
 - spielen in der Anwendung eine wichtige Rolle (Existenz sehr effizienter Parseverfahren)
 - werden in der zweiten Hälfte von FGI-1 behandelt
- inwieweit Grammatiken „vereinfacht“ werden können. Dies betrifft insbesondere die Konstruktion von Beweisen im Hinblick auf „, aber auch das Entwerfen und Realisieren von effizienten Parsern
- Normalformen, insbesondere Chomsky-Normalform: Alle Regeln haben die Form $A \rightarrow BC$ oder $A \rightarrow a$ mit $A, B, C \in N$ und $a \in \Sigma$.

Spezifische Regelformen: Einseitig lineare Grammatiken

Definition 1.11

Sei Σ ein Alphabet und $G = (\Sigma, N, P, S)$ eine kontextfreie Grammatik über Σ .

- G heißt genau dann **rechtslinear**, wenn $P \subseteq N \times (\Sigma^*N \cup \Sigma^*)$.
- G heißt genau dann **linkslinear**, wenn $P \subseteq N \times (N\Sigma^* \cup \Sigma^*)$.
- G heißt genau dann **einseitig linear**, wenn G rechtslinear oder linkslinear ist.

Anmerkung

- ‚Linearität‘ bezieht sich jeweils darauf, dass in jeder zwischenzeitlich erzeugten Satzform maximal ein Nichtterminalsymbol auftritt.
- ‚Einseitig‘ besagt zudem, dass das Nichtterminalsymbol randständig sein und bleiben muss.
- Alle Regeln der einseitig linearen Grammatik haben das Nichtterminalsymbol auf derselben Seite.

Beispiel: Lineare Grammatiken

Rechtslinear

$$G_{rl} = (\{0, 1\}, \{S\}, P_{rl}, S) \text{ mit } P_{rl} = \{ S \rightarrow 0S, S \rightarrow 1S, S \rightarrow 100 \}$$

Linkslinear

$$G_{ll} = (\{0, 1\}, \{S, R\}, P_{ll}, S) \text{ mit } P_{ll} = \{ S \rightarrow R100, R \rightarrow R1, R \rightarrow R0, R \rightarrow \varepsilon \}$$

Linear aber nicht einseitig linear

$$G_l = (\{a, b\}, \{S\}, P_l, S) \text{ mit } P_l = \{ S \rightarrow aSb, S \rightarrow ab \}$$

$$L(G_l) = \{ a^n b^n \mid n \geq 1 \}$$

(siehe auch Folie 1-16)

Zwischenfazit: Grammatiken, Ableitungen & Strukturbäume

- Kontextfreie Grammatiken erzeugen
 - Sprachen, d.h. Mengen von Zeichenketten
 - Syntaktische Strukturen zu diesen Zeichenketten
- Syntaktische Strukturen von Zeichenketten spielen bei der Zuweisung von Bedeutung eine zentrale Rolle
 - Deswegen sind Maßnahmen zur Herbeiführung von Eindeutigkeit, d.h. Maßnahmen zur Vermeidung von Mehrdeutigkeit (Ambiguität) wichtig.
- Effiziente Parsingverfahren, d.h. Verfahren zur Berechnung von Strukturbäumen zu Zeichenketten, sind für die Auswertung von Zeichenketten von grosser Relevanz. Dies betrifft:
 - Parsing von Programmiersprachen und anderen Sprachen zur Interaktion mit Systemen (z.B. Datenbanken & Informationssystemen)
 - maschinelle Verarbeitung natürlicher Sprache (u.a. Information-retrieval, maschinelle Übersetzung, Mensch-Computer-Interaktion)

➔ Weiterführung des Themas: Grammatiken, Sprachen & Automaten in der zweiten Hälfte von FGI-1

Literaturhinweis

Weiteres Material und weitere Beispiele zu den Definitionen und Charakterisierungen dieses Kapitels finden Sie bei

Vossen, Gottfried & Witt, Kurt-Ulrich (2006). Grundkurs Theoretische Informatik. Vieweg Verlag.

- Zu Alphabeten, Zeichenketten & Sprachen: Kap. 2.1.1
- Kontextfreie Grammatiken & Sprachen: Kap. 5.1

Das Thema Kontextfreie Grammatiken & Sprachen wird im weiteren Verlauf von FGI-1 noch vertieft behandelt werden.