

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 unten 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 Ziffernmeng: $B = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

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

- 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.

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 .
- 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 $u \stackrel{*}{\Rightarrow} v$ notiert.

Derartige Ableitungen sind stets von endlicher Länge (endliche Anzahl von Schritten); die Ableitungssequenz kann aber beliebige Länge haben.

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

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

Einige Ableitungen:

$$S \Rightarrow ab \qquad S \Rightarrow aSb \Rightarrow aabb \qquad S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$$

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

Beweis als Nachbereitungsaufgabe.

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

$G_2 = (\{a, b\}, \{S\}, P, S)$ mit
 $P = \{ S \rightarrow aSb, S \rightarrow \varepsilon \}$
 $S \Rightarrow \varepsilon \qquad S \Rightarrow aSb \Rightarrow ab \qquad S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$
 $L(G_2) = \{ a^n b^n \mid n \geq 0 \}, \quad \text{also } \varepsilon \in L(G_2)$

Strukturbäume (Parse trees)

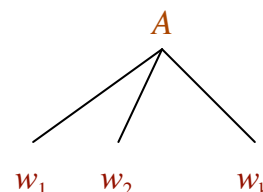
Sei $A \rightarrow w$ eine Regel, mit $|w| = k$,

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 .



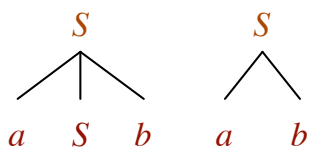
Sei $S \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \dots \Rightarrow u_k \Rightarrow w$ die Ableitung

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

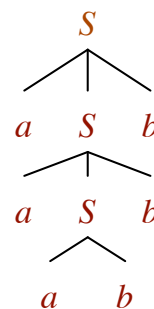
mit $P = \{ S \rightarrow aSb, S \rightarrow ab \}$



$S \Rightarrow aSb$

$\Rightarrow aaSbb$

$\Rightarrow aaabbb$



Beispiel: 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.

Nocheinmal: Arithmetische Ausdrücke

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

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

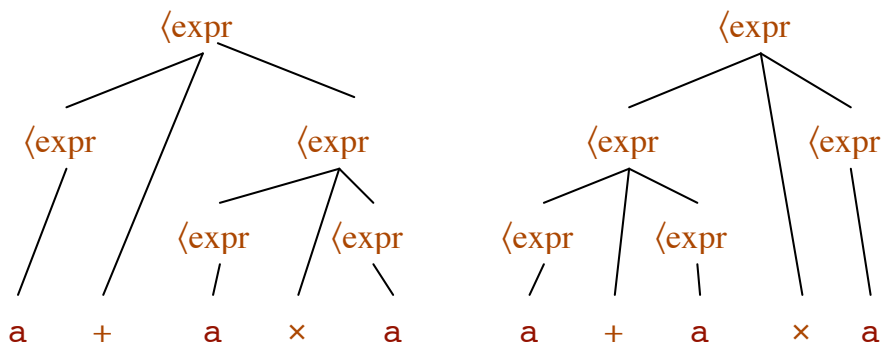
$$N = \{ \langle \text{EXPR} \rangle \}$$

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

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

Nachprüfen!!

Aber $a + a \times a$ besitzt zwei Ableitungen, mit verschiedenen Strukturbäumen.



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. Informationretrieval)

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.