Definition 4 (Operationen auf Sprachen)

Seien $A, B \subseteq \Sigma^*$ zwei (formale) Sprachen.

- Konkatenation: $AB = \{uv\,;\, u \in A, v \in B\}$
- $A^0 = \{\epsilon\}$, $A^{n+1} = AA^n$
- $A^* = \bigcup_{n \geq 0} A^n$
- $A^+ = \bigcup_{n \geq 1} A^n$

Beispiel 5

- $\{ab, b\}\{a, bb\} = \{aba, abbb, ba, bbb\}$
  $\{ab, b\} \times \{a, bb\} = \{(ab, a), (ab, bb), (b, a), (b, bb)\}$
- $\{ab, b\}^2 = \{abab, abb, bab, bb\}$
- $\{ab, a\}\{ba, a\} = \{abba, aba, aa\}$
- $\emptyset^* = \{\epsilon\}$

Einige nützliche Rechenregeln:

- $\emptyset A = A\emptyset = \emptyset$
- $\{\epsilon\}A = A\{\epsilon\} = A$
- $A(B \cup C) = AB \cup AC$
- $(A \cup B)C = AC \cup BC$
- $A(B \cap C) = AB \cap AC$ gilt i.A. nicht!
- $A^*A^* = A^*$

Wie bekannt heißt eine Menge M abzählbar, falls sie gleich mächtig wie eine Teilmenge der natürlichen Zahlen $\mathbb{N}$ (bzw. $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$) ist, d.h., falls eine Bijektion zwischen den beiden Mengen existiert.
Dies ist auch gleichbedeutend mit der Aussage, dass es eine Nummerierung der Elemente von M gibt, so dass

$$M = \{m_1, m_2, \ldots\}\,.$$

Eine Menge heißt überabzählbar, falls sie nicht abzählbar ist.

Lemma 6

Für endliches Σ ist Σ^* abzählbar.

Beweis:

Ohne Beweis. □

Bemerkungen:

- $\mathbb{Q}$ ist abzählbar.
- $\mathbb{R}$ und $[0,1] \subset \mathbb{R}$ sind gleich mächtig (haben gleiche Kardinalität) und sind beide überabzählbar.

Satz 7

Die Menge der Sprachen über einem (nichtleeren) endlichen Alphabet ist überabzählbar.

Beweis:

Widerspruchsbeweis durch Diagonalisierung: Angenommen, $L_0, L_1, L_2, \ldots$ sei eine Nummerierung der Sprachen über Σ. Sei weiter $w_0, w_1, w_2, \ldots$ eine (feste) Abzählung von Σ^*.
Betrachte $L := \{w_i; w_i \notin L_i\}$. Dann kann L nicht in der Nummerierung $L_0, L_1, L_2, \ldots$ vorkommen! $\square$

2. Die Chomsky-Hierarchie

Diese Sprachenhierarchie ist nach Noam Chomsky [MIT, 1976] benannt.

2.1 Phrasenstrukturgrammatik, Chomsky-Grammatik

Grammatiken bestehen aus

1. einem Terminalalphabet Σ (manchmal auch T), $|\Sigma| < \infty$
2. einem endlichen Vorrat von Nichtterminalzeichen (Variablen) V, $V \cap \Sigma = \emptyset$
3. einem Startsymbol (Axiom) $S \in V$
4. einer endliche Menge P von Produktionen (Ableitungsregeln) der Form $l \to r$, mit $l \in (V \cup \Sigma)^* V (V \cup \Sigma)^*$, $r \in (V \cup \Sigma)^*$

Eine Phrasenstrukturgrammatik (Grammatik) ist ein Quadrupel $G = (V, \Sigma, P, S)$.

Sei $G = (V, \Sigma, P, S)$ eine Phrasenstrukturgrammatik.

Definition 8

Wir schreiben

1. $z \rightarrow_G z'$ gdw $(\exists x, y \in (V \cup \Sigma)^*, l \rightarrow r \in P)[z = xly, z' = xry]$
2. $z \rightarrow_G^* z'$ gdw $z = z'$ oder $z \rightarrow_G z^{(1)} \rightarrow_G z^{(2)} \rightarrow_G \ldots \rightarrow_G z^{(k)} = z'$. Eine solche Folge von Ableitungsschritten heißt eine Ableitung für z' von z in G (der Länge k).
3. Die von G erzeugte Sprache ist

$$L(G) := \{z \in \Sigma^*;\ S \rightarrow_G^* z\}$$

Zur Vereinfachung der Notation schreiben wir gewöhnlich $\rightarrow$ und $\rightarrow^*$ statt $\rightarrow_G$ und $\rightarrow_G^*$

Vereinbarung:
Wir bezeichnen Nichtterminale mit großen und Terminale mit kleinen Buchstaben!

Beispiel 9

Wir erinnern uns:

- $L_2 = \{ab, abab, ababab, \ldots\} = \{(ab)^n; n \in \mathbb{N}\} \qquad (\Sigma_2 = \{a, b\})$
- Grammatik für L_2 mit folgenden Produktionen:

$$S \to ab, S \to abS$$

Beispiel 9 (Forts.)

- $L_4 = \{a, b, aa, ab, bb, aaa, aab, abb, bbb \ldots\}$
  $= \{a^m b^n; m, n \in \mathbb{N}_0, m + n > 0\} \qquad (\Sigma_4 = \{a, b\})$
- Grammatik für L_4 mit folgenden Produktionen:

$$
\begin{aligned}
&S \to A,\ S \to B,\ S \to AB,\\
&A \to a,\ A \to aA,\\
&B \to b,\ B \to bB
\end{aligned}
$$

2.2 Die Chomsky-Hierarchie

Sei $G = (V, \Sigma, P, S)$ eine Phrasenstrukturgrammatik.

1. Jede Phrasenstrukturgrammatik (Chomsky-Grammatik) ist (zunächst) automatisch vom Typ 0.
2. Eine Chomsky-Grammatik heißt (längen-)monoton, falls für alle Regeln

$$\alpha \to \beta \in P \text{ mit } \alpha \neq S$$

gilt:

$$|\alpha| \leq |\beta| \, ,$$

und, falls $S \to \epsilon \in P$, dann das Axiom S auf keiner rechten Seite vorkommt.

3. Eine Chomsky-Grammatik ist vom Typ 1 (auch: kontextsensitiv), falls sie monoton ist und für alle Regeln $\alpha \to \beta$ in P mit $\alpha \neq S$ gilt:

$$\alpha = \alpha' A \alpha'' \text{ und } \beta = \alpha' \beta' \alpha''$$

für geeignete $A \in V$, $\alpha', \alpha'' \in (V \cup \Sigma)^*$ und $\beta' \in (V \cup \Sigma)^+$.

4. Eine Chomsky-Grammatik ist vom Typ 2 (auch: kontextfrei), falls sie monoton ist und für alle Regeln $\alpha \to \beta \in P$ gilt:

$$\alpha \in V\ .$$

Bemerkung: Manchmal wird "kontextfrei" auch ohne die Monotonie-Bedingung definiert; streng monoton schließt dann die Monotonie mit ein, so dass ϵ nicht als rechte Seite vorkommen kann.

5. Eine Chomsky-Grammatik ist vom Typ 3 (auch: regulär, rechtslinear), falls sie monoton ist und für alle Regeln $\alpha \to \beta$ in P mit $\beta \neq \epsilon$ gilt:

$$\alpha \in V \text{ und } \beta \in \Sigma^+ \cup \Sigma^* V \ .$$

Auch hier gilt die entsprechende Bemerkung zur Monotonie-Bedingung.

Beispiel 10

- Die folgende Grammatik ist regulär:

$$S \to \epsilon,\ S \to A,$$
$$A \to aa,\ A \to aaA$$

- Eine Produktion

$$A \to Bcde$$

heißt linkslinear.

- Eine Produktion

$$A \to abcDef$$

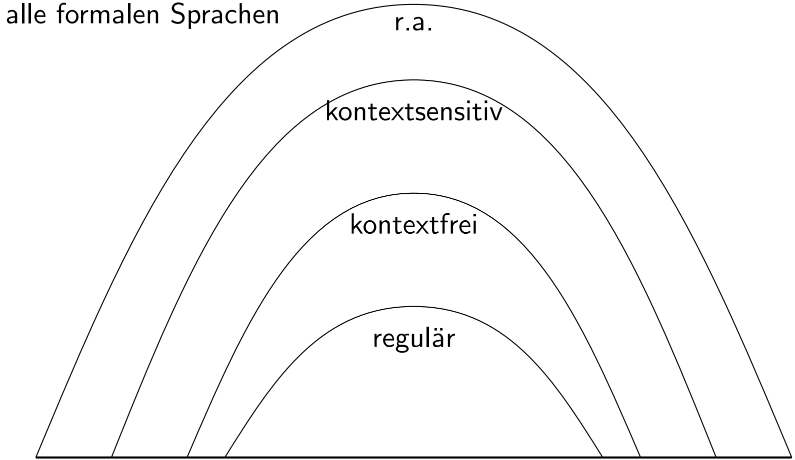
heißt linear.

Definition 11

Eine Sprache $L \subseteq \Sigma^*$ heißt vom Typ k, $k \in \{0, 1, 2, 3\}$, falls es eine Chomsky-k-Grammatik G mit $L(G) = L$ gibt.

In der Chomsky-Hierarchie bilden also die Typ-3- oder regulären Sprachen die kleinste, unterste Stufe, darüber kommen die kontextfreien, dann die kontextsensitiven Sprachen. Oberhalb der Typ-1-Sprachen kommen die Typ-0-Sprachen, die auch rekursiv aufzählbar oder semi-entscheidbar genannt werden. Darüber (und nicht mehr Teil der Chomsky-Hierarchie) findet sich z.B. die Klasse aller formalen Sprachen.

In Typ-3-Grammatiken müssen entweder alle Produktionen rechtslinear oder alle linkslinear sein.
Überlegen Sie sich eine lineare Grammatik, deren Sprache nicht regulär ist!
(Beweismethode später!)


alle formalen Sprachen
r.a.
kontextsensitiv
kontextfrei
regulär

Lemma 12

Sei $G = (V, \Sigma, P, S)$ eine Chomsky-Grammatik, so dass alle Produktionen $\alpha \to \beta$ die Bedingung $\alpha \in V$ erfüllen. Dann ist $L(G)$ kontextfrei.

Beweis:

Definition 13

Ein $A \in V$ mit $A \to^* \epsilon$ heißt nullierbar.

Bestimme alle nullierbaren $A \in V$:

$N := \{A \in V;\ (A \to \epsilon) \in P\}$
$N' := \emptyset$
while $N \neq N'$ **do**
  $N' := N$
  $N := N' \cup \{A \in V;$
    $(\exists(A \to \beta) \in P)[\beta \in N'^*]\}$
od

Wie man leicht durch Induktion sieht, enthält N zum Schluss genau alle nullierbaren $A \in V$.

Sei nun G eine Grammatik, so dass alle linken Seiten $\in V$, aber die Monotoniebedingung nicht unbedingt erfüllt ist.
Modifiziere G zu G' mit Regelmenge P' wie folgt:

1. für jedes $(A \to x_1 x_2 \cdots x_n) \in P$, $n \geq 1$, füge zu P' alle Regeln $A \to y_1 y_2 \cdots y_n$ hinzu, die dadurch entstehen, dass für nicht-nullierbare x_i $y_i := x_i$ und für nullierbare x_i die beiden Möglichkeiten $y_i := x_i$ und $y_i := \epsilon$ eingesetzt werden, ohne dass die ganze rechte Seite $= \epsilon$ wird.
2. falls S nullierbar ist, sei T ein neues Nichtterminal; füge zu P' die Regeln $S \to \epsilon$ und $S \to T$ hinzu, ersetze S in allen rechten Seiten durch T und ersetze jede Regel $(S \to x) \in P'$, $|x| > 0$, durch $T \to x$.

Lemma 14

$G' = (V \cup T, \Sigma, P', S)$ ist kontextfrei, und es gilt

$$L(G') = L(G)\ .$$

Beweis:

Klar! $\square$

Auch für reguläre Grammatiken gilt ein entsprechender Satz über die "Entfernbarkeit" nullierbarer Nichtterminale:

Lemma 15

Sei $G = (V, \Sigma, P, S)$ eine Chomsky-Grammatik, so dass für alle Regeln $\alpha \to \beta \in P$ gilt:

$$\alpha \in V \text{ und } \beta \in \Sigma^* \cup \Sigma^* V \ .$$

Dann ist $L(G)$ regulär.

Beweis:

Übungsaufgabe! $\square$

Beispiel 16

Typ 3: $L = \{a^n;\ n \in \mathbb{N}\}$, Grammatik:

$$\begin{aligned} S &\rightarrow a, \\ S &\rightarrow aS \end{aligned}$$

Typ 2: $L = \{a^n b^n;\ n \in \mathbb{N}_0\}$, Grammatik:

$$\begin{aligned} S &\rightarrow \epsilon, \\ S &\rightarrow T, \\ T &\rightarrow ab, \\ T &\rightarrow aTb \end{aligned}$$

Wir benötigen beim Scannen *einen* Zähler.

Beispiel 16 (Forts.)

Typ 1: $L = \{a^n b^n c^n;\ n \in \mathbb{N}\}$, Grammatik:

$$
\begin{array}{rcl}
S & \rightarrow & aSXY\,, \\
S & \rightarrow & abY\,, \\
YX & \rightarrow & XY\,, \\
bX & \rightarrow & bb, \\
bY & \rightarrow & bc, \\
cY & \rightarrow & cc
\end{array}
$$

Wir benötigen beim Scannen *mindestens zwei* Zähler.

Bemerkung: Diese Grammatik entspricht *nicht* unserer Definition des Typs 1, sie ist aber (längen-)monoton. Wir zeigen als Hausaufgabe, dass monotone und Typ 1 Grammatiken die gleiche Sprachklasse erzeugen!

Die Backus-Naur-Form (BNF) ist ein Formalismus zur kompakten Darstellung von Typ-2-Grammatiken.

- Statt

$$\begin{array}{lcl} A & \to & \beta_1 \\ A & \to & \beta_2 \\ & \vdots & \\ A & \to & \beta_n \end{array}$$

schreibt man

$$A \to \beta_1 | \beta_2 | \ldots | \beta_n \; .$$

Die Backus-Naur-Form (BNF) ist ein Formalismus zur kompakten Darstellung von Typ-2-Grammatiken.

- Statt

$$\begin{array}{rcl} A & \rightarrow & \alpha\gamma \\ A & \rightarrow & \alpha\beta\gamma \end{array}$$

  schreibt man

$$A \rightarrow \alpha[\beta]\gamma\ .$$

(D.h., das Wort β kann, muss aber nicht, zwischen α und γ eingefügt werden.)

Die Backus-Naur-Form (BNF) ist ein Formalismus zur kompakten Darstellung von Typ-2-Grammatiken.

- Statt

$$
\begin{array}{lcl}
A & \rightarrow & \alpha\gamma \\
A & \rightarrow & \alpha B\gamma \\
B & \rightarrow & \beta \\
B & \rightarrow & \beta B
\end{array}
$$

  schreibt man

$$A \rightarrow \alpha\{\beta\}\gamma\ .$$

  (D.h., das Wort β kann beliebig oft (auch Null mal) zwischen α und γ eingefügt werden.)

Beispiel 17

$$
\begin{array}{rcl}
\langle\text{Satz}\rangle & \rightarrow & \langle\text{Subjekt}\rangle\langle\text{Prädikat}\rangle\langle\text{Objekt}\rangle \\
\langle\text{Subjekt}\rangle & \rightarrow & \langle\text{Artikel}\rangle\langle\text{Attribut}\rangle\langle\text{Substantiv}\rangle \\
\langle\text{Prädikat}\rangle & \rightarrow & \text{ist}|\text{hat}|\ldots \\
\langle\text{Artikel}\rangle & \rightarrow & \epsilon|\text{der}|\text{die}|\text{das}|\text{ein}|\ldots \\
\langle\text{Attribut}\rangle & \rightarrow & \{\langle\text{Adjektiv}\rangle\} \\
\langle\text{Adjektiv}\rangle & \rightarrow & \text{gross}|\text{klein}|\text{schön}|\ldots \\
\langle\text{Substantiv}\rangle & \rightarrow & \ldots
\end{array}
$$

2.3 Das Wortproblem

Beispiel 18 (Arithmetische Ausdrücke)

$$
\begin{array}{lcl}
\langle\text{expr}\rangle & \to & \langle\text{term}\rangle \\
\langle\text{expr}\rangle & \to & \langle\text{expr}\rangle + \langle\text{term}\rangle \\
\langle\text{term}\rangle & \to & (\langle\text{expr}\rangle) \\
\langle\text{term}\rangle & \to & \langle\text{term}\rangle \times \langle\text{term}\rangle \\
\langle\text{term}\rangle & \to & a \mid b \mid \ldots \mid z
\end{array}
$$

Aufgabe eines Parsers ist nun, zu prüfen, ob eine gegebene Zeichenreihe einen gültigen arithmetischen Ausdruck darstellt und, falls ja, ihn in seine Bestandteile zu zerlegen.

Sei $G = (V, \Sigma, P, S)$ eine Grammatik.

Definition 19

1. **Wortproblem:** Gegeben ein Wort $w \in \Sigma^*$, stelle fest, ob
$$w \in L(G)\ ?$$
2. **Ableitungsproblem:** Gegeben ein Wort $w \in L(G)$, gib eine Ableitung $S \rightarrow_G^* w$ an, d.h. eine Folge
$$S = w^{(0)} \rightarrow_G w^{(1)} \rightarrow_G \cdots \rightarrow_G w^{(n)} = w$$
mit $w^{(i)} \in (\Sigma \cup V)^*$ für $i = 1, \ldots, n$.
3. **uniformes Wortproblem:** Wortproblem, bei dem jede Probleminstanz sowohl die Grammatik G wie auch die zu testende Zeichenreihe w enthält. Ist G dagegen global festgelegt, spricht man von einem nicht-uniformen Wortproblem.