4. Das Wortproblem

4.1 Die Existenz von Ableitungen eines Wortes

Beispiel 56 (Arithmetische Ausdrücke)

$$
\begin{array}{lcl}
<\text{expr}> & \rightarrow & <\text{term}> \\
<\text{expr}> & \rightarrow & <\text{expr}> + <\text{term}> \\
<\text{term}> & \rightarrow & (<\text{expr}>) \\
<\text{term}> & \rightarrow & <\text{term}> \times <\text{term}> \\
<\text{term}> & \rightarrow & a \mid b \mid \ldots \mid z
\end{array}
$$

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 57

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.

Bemerkung:
Das uniforme wie auch das nicht-uniforme Wortproblem ist für Typ-0-Sprachen (also die rekursiv-aufzählbare Sprachen) nicht entscheidbar. Wir werden später sehen, dass es zum Halteproblem für Turingmaschinen äquivalent ist.

Es gilt jedoch

Satz 58

Für kontextsensitive Grammatiken ist das Wortproblem entscheidbar.
Genauer: Es gibt einen Algorithmus, der bei Eingabe einer kontextsensitiven Grammatik $G = (V, \Sigma, P, S)$ und eines Wortes w in endlicher Zeit entscheidet, ob $w \in L(G)$.

Beweisidee:
Angenommen $w \in L(G)$. Dann gibt es eine Ableitung

$$S = w^{(0)} \rightarrow_G w^{(1)} \rightarrow_G \cdots \rightarrow_G w^{(n)} = w$$

mit $w^{(i)} \in (\Sigma \cup V)^*$ für $i = 1, \ldots, n$.
Da aber G kontextsensitiv ist, gilt (falls $w \neq \epsilon$)

$$|w^{(0)}| \leq |w^{(1)}| \leq \cdots \leq |w^{(n)}| \,,$$

d.h., es genügt, alle Wörter in $(\Sigma \cup V)^*$ der Länge $\leq |w|$ zu erzeugen.

Beweis:

Sei o.B.d.A. $w \neq \epsilon$ und sei

$$T_m^n \quad := \quad \{w' \in (\Sigma \cup V)^*;\ |w'| \leq n \text{ und } w' \text{ lässt sich aus } S \text{ in } \leq m \text{ Schritten ableiten}\}$$

Diese Mengen kann man für alle n und m induktiv wie folgt berechnen:

$$\begin{aligned} T_0^n &:= \{S\} \\ T_{m+1}^n &:= T_m^n \cup \{w' \in (\Sigma \cup V)^*;\ |w'| \leq n \text{ und } \\ &\qquad\qquad w'' \to w' \text{ für ein } w'' \in T_m^n\} \end{aligned}$$

Beachte: Für alle m gilt: $|T_m^n| \leq \sum_{i=1}^n |\Sigma \cup V|^i$.
Es muss daher immer ein m_0 geben mit

$$T_{m_0}^n = T_{m_0+1}^n = \ldots =: T_n\ .$$

Beweis (Forts.):

Algorithmus:

$n := |w|$
$T := \{S\}$
$T' := \emptyset$
while $T \neq T'$ **do**
  $T' := T$
  $T := T' \cup \{w' \in (V \cup \Sigma)^+;\ |w'| \leq n, (\exists w'' \in T')[w'' \rightarrow w']\}$
od
if $w \in T$ **return** „ja“ **else** **return** „nein“ **fi**

$\square$

Beispiel 59

Gegeben sei die Typ-2-Grammatik mit den Produktionen

$$S \to ab \text{ und } S \to aSb$$

sowie das Wort $w = abab$.

$$\begin{aligned}
T_0^4 &= \{S\} \\
T_1^4 &= \{S, ab, aSb\} \\
T_2^4 &= \{S, ab, aSb, aabb\} \quad aaSbb \text{ ist zu lang!} \\
T_3^4 &= \{S, ab, aSb, aabb\}
\end{aligned}$$

Also lässt sich das Wort w mit der gegebenen Grammatik nicht erzeugen!

Bemerkung:
Der angegebene Algorithmus ist nicht sehr effizient! Für kontextfreie Grammatiken gibt es wesentlich effizientere Verfahren, die wir später kennenlernen werden!