noch 2.1.1 Grammatiken

Notation: Σ^*, $\Sigma^+ = \Sigma^* - \{\varepsilon\}$

Def (Grammatik)

Eine Grammatik ist ein 4-Tupel $G = (V, \Sigma, P, S)$,

wobei

- V eine endl. Menge von Variablensymbolen ist,
- Σ das Terminalalphabet mit $\Sigma \cap V = \emptyset$ ist,
- $P \subseteq (V \cup \Sigma)^+ \times (V \cup \Sigma)^*$ die Menge der Produktionsregeln ist, und
- $S \in V$ ist das Startsymbol.

Bsp

$H = (V, \Sigma, P, S)$

$V = \{S\}$ $\Sigma = \{a, b\}$, $P = \{S \to \varepsilon, S \to aSb\}$

Notation: Oft wird nur P angegeben (Variablen: Großbuchstaben, Terminalsymb.: Kleinbuchstaben, S ist immer Startsymb.

Def (Ableitungen)
Sei $G = (V, \Sigma, P, S)$ Gram. und $v, w \in (V \cup \Sigma)^*$. Dann kann v in einem Schritt von aus v in G abgeleitet werden, $v \Rightarrow_G w$, falls
$v = xyz$, $w = xy'z$, $x, z \in (V \cup \Sigma)^*$, und $y \to y' \in P$.
$\Rightarrow_G^*$ ist die reflexive, transitive Hülle von $\Rightarrow_G$.
Die Folge $(w_0, w_1, \dots, w_n)$ mit $w_0 = S$, $w_n \in \Sigma^*$, $w_i \in (V \cup \Sigma)^*$ heißt Ableitung von w_n in G falls
$w_0 \Rightarrow_G w_1 \Rightarrow_G \dots \Rightarrow_G w_n$.

Notation: Oft wird der Subskript G weggelassen

Bsp:

$$S \Rightarrow_H aSb \Rightarrow_H aaSbb \Rightarrow_H aabb$$

Def (Erzeugte Sprache)

Die von $G = (V, \Sigma, P, S)$ erzeugte Sprache $L(G)$ ist def.:

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

Bsp:

$$L(H) = \{ a^n b^n \mid n \geq 0 \}$$

Bsp $\quad G = (\{S, B, C\}, \{a, b, c\}, P, S)$

$P = \{ S \to aSBC,\ S \to aBC,\ CB \to BC,\ aB \to ab,\ bB \to bb,\ bC \to bc,\ cC \to cc \}$

$S \Rightarrow aSBC \Rightarrow aaBCBC \Rightarrow aabCBC \Rightarrow aabcBC \Rightarrow aabBC$

$\Rightarrow aaBBCC \Rightarrow aabBCC$

$\Rightarrow aabbCC \Rightarrow aabbcC \Rightarrow aabbcc$

Beh: $L(G) = \{ a^n b^n c^n \mid n \geq 1 \}$

($\Rightarrow$) $S \Rightarrow^* a^{n-1} S (BC)^{n-1}$ Regel n-1 anwenden

$\Rightarrow a^n (BC)^n$ Regel 2 1x anwende

$\Rightarrow^* a^n B^n C^n$ Regel 3 so oft wie möglich

$\Rightarrow a^n b B^{n-1} C^n$ Regel 4

$\Rightarrow^* a^n b^n C^n$ Regel 5 n-1 x

$\Rightarrow a^n b^n c C^{n-1}$ Regel 6

$\Rightarrow^* a^n b^n c^n$ Regel 7 n-1 x

($\subseteq$): 1. In jedem Ableitungsschritt ist die Anzahl der a's (Groß oder klein) gleich der Anzahl der b's (Groß oder klein) gleich der Anzahl der c's (groß oder klein).
2. a's können nur ganz links stehen (Regel 1,2 und 4)
3. Das Teilwort aus b's und c's wird aus den Regeln 4-7 erzeugt. Steht B rechts von c, dann kann daraus kein b werden. Alle b's müssen links von den kleinen c's stehen. □

2.12 Chomsky-Hierarchie

Def (Chomsky-N.)

Jede Gram. ist vom Typ 0.
Eine Gram. ist vom Typ 1 (kontextsensitiv) falls für alle $w_1 \to w_2 \in P$ gilt: $|w_1| \leq |w_2|$,
Eine Gram. ist vom Typ 2 wenn ~~zusätzlich~~ für alle $w_1 \to w_2 \in P$ gilt, dass $w_1 \in V$. (kontextfreie Sprachen).

Eine Typ 2-Gram. ist vom Typ 3 (reguläre) falls für alle $w_1 \to w_2 \in P$ gilt $w_2 \in \Sigma \circ \Sigma V$.

Eine Spr. ist vom Typ i falls es eine Gram. vom Typ i gibt, die sie erzeugt.

Bsp. F ist eine Typ 1-Grammatik, d.h. $L(F)$ ist eine Typ 1-Spr.

H ist eine Typ 0-Gram. / keine Typ 1, keine Typ 2-Gram., $L(H)$ ist eine Typ 2-Sprache.

Alternativ zu H:

$H' = (V, \Sigma, P', S)$ $P' = \{S \to ab, S \to aSb\}$ $L(H') = \{a^n b^n \mid n \geq 1\}$

ε-Sonderregel: Wg. $|w_1| \leq |w_2|$ kann eigentlich ε nie ein Wort einer Typ 1-Sprache sein. Deshalb ist die Regel $S \to \varepsilon$ erlaubt, falls S nicht auf der rechten Seite auftaucht.

Bsp:

$G'' = (V, \Sigma, P'', S)$

$V = \{S, S'\}$,

$P'' = \{S \to \varepsilon,\ S \to S',\ S' \to ab,\ S' \to aS'b\}$

$\{\varepsilon\}$ $\qquad$ $\{a^n b^n \mid n \geq 1\}$ $\qquad$ $= \{a^n b^n \mid n \geq 0\}$

Bem: Oft will man in einer Typ 2/Typ 3 Gramm. die Regel $\frac{A \to \varepsilon}{}$ zulassen. Für jede Gramm G mit $\varepsilon \notin L(G)$ können wir G' erzeugen ohne ε-Regeln mit $L(G) = L(G')$ (ansonsten ε-Sonderregel anwende), Bestimme alle $A \in V$ mit $A \overset{*}{\Longrightarrow}_G \varepsilon$. Dann erzeuge für jede Regel $B \to xAy$ $(x, y \in (V \cup \Sigma)^*)$, die Regel $B \to xy$. Lösche alle ε-Regeln.

Bsp

$S \to (A)$

$A \to BC$

$B \to \varepsilon$

$B \to b$

$C \to c$

$C \to \varepsilon$

$B \Rightarrow^* \varepsilon$

$C \Rightarrow^* \varepsilon$

$A \Rightarrow^* \varepsilon$

$S \to (A)$

$S \to ()$

$A \to BC$

$A \to B$

$A \to C$

$B \to b$

$C \to c$

$L = L(H) = \{ a^n b^n \mid n \geq 0 \}$ ist Typ 2 aber nicht Typ 3

$L' = L(F) = \{ a^n b^n c^n \mid n \geq 1 \}$ ist Typ 1 aber nicht Typ 2

$L'' = \{ \varphi \mid \varphi$ ist eine allgemeingültige Formel der Prädikatenlogik 1. Stufe $\}$ ist Typ 0, aber nicht Typ 1

$\overline{L''}$ ist nicht Typ 0

Bem: kontextfreie / reguläre Spr. werden gern benutzt um Programmiersprachen zu beschreiben; allerdings können nicht alle Aspekte von Prog.-Spr. so zu beschreiben.

2.1.3 Wortproblem

Def (Wortproblem)

Gegeben eine Gram. $G = (V, \Sigma, P, S)$ und ein Wort $w \in \Sigma^*$, dann ist das **Wortproblem** zu entscheiden ob $w \in L(G)$.

Satz 2

Es existiert ein Algorithmus, der das Wortproblem für Typ 1 Spr. entscheidet.

31 Okt = 25 Dez