3.4 Der CYK-Algorithmus

Ziel: effizienter Algorithmus, mit dessen Hilfe bei Eingabe eines Wortes w entschieden werden kann, ob es von der kontextfreien Grammatik G erzeugt wird.
1. Idee: probiere aus, eine Ableitung zu konstruieren, die mit S beginnt und mit w endet

1. Aber dann ist nicht klar, wann man aufgeben soll. (Aufgrund der leeren rechten Seiten kann es sehr lange Ableitungen mit sehr kurzem Ergebniswort geben).

   Bsp.: $S \to A_1A_1$, $A_i \to A_{i+1}A_{i+1}$ für alle $1 \leq i < n$, $A_n \to \varepsilon$.
   Dann gilt $S \Rightarrow^* \varepsilon$, aber die einzige Ableitung hat die Länge $2^{n+1} - 1$.

2. selbst wenn es keine leeren rechten Seiten gibt, gibt es noch immer exponentiell viele auszuprobierende Ableitungen.

   Bsp.: $S \to A_1 \mid B_1$, $A_i \to A_{i+1} \mid B_{i+1}$ und $B_i \to A_{i+1} \mid B_{i+1}$ für alle $1 \leq i < n$, $A_n \to a$, $B_n \to a$.
   Es gibt 2^n viele Ableitungen, die alle die Länge $n + 1$ haben.

2. Idee: versuche, die Teilwörter systematisch zu erzeugen.

Sei $G = (N, \Sigma, S, P)$ eine kontextfreie Grammatik und $w = a_1 a_2 \cdots a_n \in \Sigma^+$.

Für $1 \leq \ell \leq n$ und $1 \leq i \leq n - \ell + 1$ sei

$$U_{i,\ell} = \{X \in N \mid X \Rightarrow_G^* a_i a_{i+1} \cdots a_{i+\ell-1}\}$$

die Menge der Nichtterminale, aus denen das Teilwort der Länge ℓ, das an der Stelle i beginnt, abgeleitet werden kann.

Beispiel 3.6. $w = aabcbc$ und G hat die folgenden Regeln:

$S \to AD \mid FG$	$A \to a$	$B \to b$	$C \to c$
$D \to SE \mid BC$	$E \to BC$	$F \to AF \mid a$	$G \to BG \mid CG \mid b$

Dann gelten

- $U_{11} = U_{21} = \{A, F\}$, $U_{31} = U_{51} = \{B, G\}$, $U_{41} = U_{61} = \{C\}$
- $U_{12} = \{F\}$, $U_{22} = \{S\}$, $U_{32} = \{D, E\}$, $U_{42} = \{G\}$, $U_{52} = \{D, E\}$
- $\cdots$

Es gilt offensichtlich $w \in L(G) \iff S \in U_{1,n}$, d.h. wir müssen die Menge $U_{1,n}$ berechnen.

Hierzu werden wir zunächst alle Mengen $U_{i,1}$ berechnen, dann alle Menge $U_{i,2}$ usw.

Bei der Berechnung von $U_{i,\ell}$ müssen wir also für jedes $X \in N$ herausfinden, ob $X \in U_{i,\ell}$ gilt. Dazu werden wir *nicht* versuchen, aus X das Wort $a_i a_{i+1} \cdots a_{i+\ell-1}$ abzuleiten. Vielmehr werden wir die Mengen $U_{j,k}$ mit $k < \ell$ verwenden. D.h. wir verwenden das Verfahren der *dynamischen Programmierung*.

Dies funktioniert einfacher, wenn wir eine „schöne“ kontextfreie Grammatik verwenden.

Definition. Eine kontextfreie Grammatik $G = (N, \Sigma, S, P)$ ist *in Chomsky-Normalform*, wenn gelten:

- wenn es die Regeln $S \to \varepsilon$ gibt, so kommt S auf keiner rechten Seite vor
- darüber hinaus gibt es nur Regeln der Form $A \to a$ und $A \to BC$ mit $A, B, C \in N$ und $a \in \Sigma$.

Beobachtung. Sei $G = (N, \Sigma, S, P)$ kontextfreie Grammatik in Chomsky-Normalform, $X \in N$ und $w = a_1 a_2 \cdots a_n \in \Sigma^+$

Dann gilt $X \in U_{1,n}$ (d.h. $X \Rightarrow_G^* w$) genau dann, wenn

- $n = 1$ und $X \to w \in P$ oder
- $n > 1$ und es existieren $1 \leq k < n$ und $X \to YZ \in P$ mit $Y \Rightarrow_G^* a_1 a_2 \cdots a_k$ und $Z \Rightarrow_G^* a_{k+1} \cdots a_n$, d.h. $Y \in U_{1,k}$ und $Z \in U_{k+1,n-(k+1)}$

Allgemeiner gilt $X \in U_{i,\ell}$ genau dann, wenn

- $\ell = 1$ und $X \to a_i \in P$ oder
- $\ell > 1$ und es gibt $X \to YZ \in P$ und $1 \leq k < \ell$ mit $Y \Rightarrow_G^* a_i a_{i+1} \cdots a_{i+k}$ und $Y \Rightarrow_G^* a_{i+k+1} a_{i+k+2} \cdots a_{i+\ell}$, d.h. $Y \in U_{i,k}$ und $Z \in U_{i+k+1,\ell-(k+1)}$

Damit kann $U_{i,\ell}$ leicht aus den Mengen $U_{j,k}$ mit $k < \ell$ berechnet werden.

Damit erhalten wir den **CYK-Algorithmus**:

Eingabe: kontextfreie Grammatik $G = (N, \Sigma, S, P)$ in Chomsky-Normalform
nichtleeres Wort $w = a_1 a_2 \cdots a_n \in \Sigma^+$

Ausgabe: „ja“, falls $w \in L(G)$, „nein“ sonst

(1) für alle $1 \leq i \leq n$ setze $U_{i,1} = \{X \in N \mid X \to a_i \in P\}$

(2) für alle $2 \leq \ell \leq n$

(3) für alle $1 \leq i \leq n - \ell + 1$

(4) $U_{i,\ell} = \{X \in N \mid \exists X \to YZ \in P, 1 \leq k < \ell \text{ mit } Y \in U_{i,k} \text{ und } Z \in U_{k+1,\ell-(k+1)}\}$

(5) falls $S \in U_{1,n}$, so gib „ja“ aus, sonst „nein“

Bemerkung. Der CYK-Algorithmus muß $O(n^2)$ viele Mengen $U_{i,\ell}$ berechnen.
Die Berechnung von $U_{i,\ell}$ betrachtet $O(|G|)$ viele Regeln $X \to YZ$ und $O(n)$ viele Zahlen k.

Der Algorithmus läuft also in Zeit $O(|G| \cdot |w|^3)$.

Beispiel 3.6 (Fortsetzung) $w = aabcbc$ und G hat die folgenden Regeln:

$S \to AD \mid FG$ $\quad A \to a$ $\quad B \to b$ $\quad C \to c$

$D \to SE \mid BC$ $\quad E \to BC$ $\quad F \to AF \mid a$ $\quad G \to BG \mid CG \mid b$

	a	a	b	c	b	c
$\ell = 1$	$U_{1,1}$	$U_{2,1}$	$U_{3,1}$	$U_{4,1}$	$U_{6,1}$	$U_{6,1}$
$\ell = 2$	$U_{1,2}$	$U_{2,2}$	$U_{3,2}$	$U_{4,2}$	$U_{5,2}$	
$\ell = 3$	$U_{1,3}$	$U_{2,3}$	$U_{3,3}$	$U_{4,3}$		
$\ell = 4$	$U_{1,4}$	$U_{2,4}$	$U_{3,4}$			
$\ell = 5$	$U_{1,5}$	$U_{2,5}$				
$\ell = 6$	$U_{1,6}$					

a	a	b	c	b	c

a	a	b	c	b	c
A, F	A, F	B, G	C	B, G	C
F	S	D, E	G	D, E	
S	S	G			
	S				
S	D				
S					

Wegen $S \in U_{1,6}$ gilt also $w \in L(G)$.

Satz 3.7. *Aus einer kontextfreien Grammatik G kann eine äquivalente kontextfreie Grammatik in Chomsky-Normalform berechnet werden.*

Beweis durch Beispiel:
Betrachte die kontextfreie Grammatik $G = (\{S\}, \{a, b\}, S, P)$ mit den Regeln

$$S \to \varepsilon \text{ und } S \to aSb\,.$$

Schritt 0: neues Startsymbol S':
$G_0 = (\{S, S'\}, \{a, b\}, S', P_0)$ mit den Regeln

$$S' \to S,\ S \to \varepsilon \text{ und } S \to aSb\,.$$

Schritt 1: Vereinzeln der Terminale:
$G_1 = (\{S, S', A, B\}, \{a, b\}, S', P_1)$ mit den Regeln

$$S' \to S,\ S \to \varepsilon \mid ASB,\ A \to a \text{ und } B \to b\,.$$

Schritt 2: Kürzen der rechten Seiten (durch Einführung neuer Nichtterminale):
$G_2 = (\{S, S', A, B, X\}, \{a, b\}, S', P_2)$ mit den Regeln

$$S' \to S \qquad S \to \varepsilon \mid AX \qquad A \to a \qquad B \to b$$
$$X \to SB$$

Schritt 3: Elimination von leeren rechten Seiten (durch Kontraktion von Ableitungen):
$G_3 = (\{S, S', A, B, X\}, \{a, b\}, S', P_3)$ mit den Regeln

$$S' \to S \mid \varepsilon \qquad S \to AX \qquad A \to a \qquad B \to b$$
$$X \to SB \mid B$$

Schritt 4: Elimination von Kettenregeln $A \to B$ (durch Kontraktion von Ableitungen):
$G_4 = (\{S, S', A, B, X\}, \{a, b\}, S', P_4)$ mit den Regeln

$$S' \to AX \mid \varepsilon \qquad S \to AX \qquad A \to a \qquad B \to b$$
$$X \to SB \mid b$$

□