2.3.3 Abschlusseigenschaften

<u>Satz</u>

Die kontextfreien Spr. sind abgeschlossen unter:

- Vereinigung
- Produkt
- Stern

Sie sind nicht abgeschlossen unter:

- Schnitt
- Komplement

Bew: Gegeben kontextfreie Gram. $G_1 = (V_1, \Sigma, P_1, S_1)$ und

$G_2 = (V_2, \Sigma, P_2, S_2)$, wobei $V_1 \cap V_2 = \emptyset$ und $S \notin (V_1 \cup V_2)$.

Vereinigung: $G = (V_1 \cup V_2 \cup \{S\}, \Sigma, P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}, S)$

Offensichtlich: $L(G) = L(G_1) \cup L(G_2)$

Produkt: $G = (V_1 \cup V_2 \cup \{S\}, \Sigma, P_1 \cup P_2 \cup \{S \to S_1 S_2\}, S)$

Offensichtlich: $L(G) = L(G_1) L(G_2)$

Stern: $L(G_1)^*$

$$G = (V_1 \cup \{S\}, \Sigma, P_1 \cup \{S \to \varepsilon, S \to S S_1\}, S)$$

Offensichtlich: $L(G) = L(G_1)^*$

Schnitt: $L_1 = \{a^i \underline{b^j c^j} \mid i,j > 0\}$ $L_2 = \{\underline{a^i b^i} c^j \mid i,j > 0\}$

$abbbccc \in L_1$ $aabbc \in L_2$

$L_1 \cap L_2 = \{a^i b^i c^i \mid i > 0\}$ ist nicht kontextfrei!

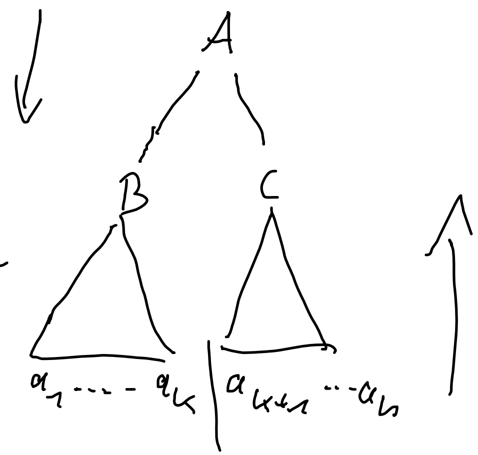
Komplement: Angenommen, die Typ2-Spr. wären unter Komplement abgeschlossen. Da $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$, müssten die Typ2 Spr. unter Schnitt abgeschlossen sein. ↯
D.h. die Typ2 Spr. können nicht unter Komplement abgeschlossen sein. □

2.3.4 CYK-Algorithmus

Effizienter Alg. für das Wortproblem kontextfreier Sprache gegeben durch CNF-Grammatiken. Autoren: Cocke, Younger, Kasami.

Idee:

Berechne Bottom-up alle möglichen Zerlegungen
→ dyn. Programmierung



Für alle Indizes, für alle Längen von Teilworten (länger werdend) berechne welches Var-Symbol das Teilwort überdeckt

Notation: Wenn x Eingabewort ist, dann soll $x_{i,j}$ das Teilwort bezeichnen, das an Pos. i beginnt und die Länge j hat

Bsp: $x = a\,b\,b\,b\,a$

$i = 1\;2\;3\;4\;5$

$x_{3,2} = bb$

$x_{1,3} = abb$

$T[i,j]$

index

Länge des Teilwortes

CYK-Algorithmus

Eingabe: $x = a_1 a_2 \ldots a_n$

FOR $i := 1$ TO n DO $(*$ Fall $j = 1 *)$
    $T[i,1] := \{A \in V \mid \boxed{A \to a_i} \in P\}$
END;
FOR $j := 2$ TO n DO $(*$ Fall $j > 1 *)$
    FOR $i := 1$ TO $n + 1 - j$ DO
        $T[i,j] := \emptyset$;
        FOR $k := 1$ TO $j - 1$ DO
            $T[i,j] := T[i,j] \cup \{A \in V \mid \boxed{A \to BC} \in P$
                $\wedge\, B \in T[i,k] \;\wedge\; C \in T[i+k, j-k]\}$
        END;
    END;
END;
IF $S \in T[1,n]$ THEN
    WriteString('x liegt in L(G)')
ELSE
    WriteString('x liegt nicht in L(G)')
END

$O(n^3)$ Laufzeit

$O(n^2)$ Speicher

Bsp: $S \to AB$

$A \to ab \mid aAb$

$B \to c \mid cB$

nach CNF

$S \to AB$

$A \to CD \mid CF$

$B \to c \mid EB$

$C \to a$ $\quad E \to c$

$D \to b$ $\quad F \to AD$

$x = aaabbbcc$

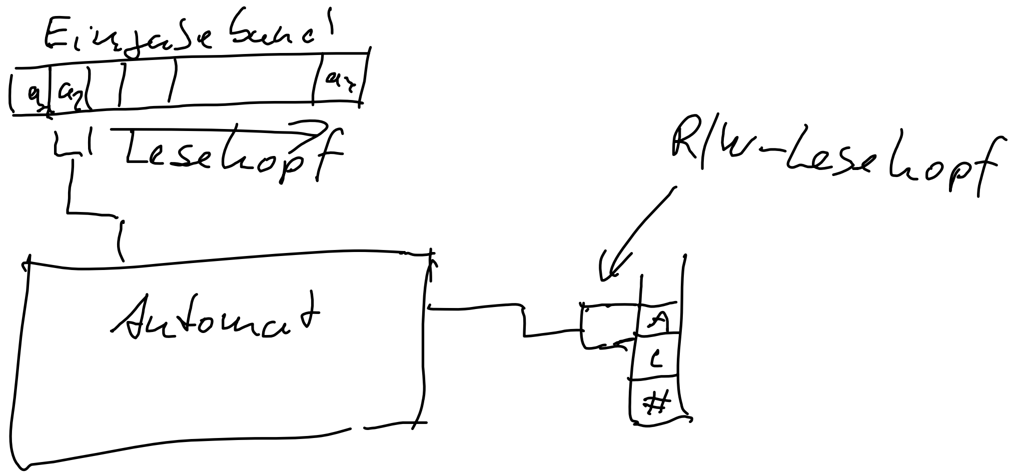
j								
8	S							
7	S							
6	A							
5		F						
4		A						
3			F					
2			A				B	
1	C	C	C	D	D	D	E, B	E, B
	a	a	a	b	b	b	c	c
$i=$	1	2	3	4	5	6	7	8

2.3.5 Kellerautomaten

$$L = \{ a^n b^n \mid n \geq 1 \}$$

Warum nicht reg? DFA hat keinen unbegrenzten Speicher.

Kellerautomat:



<u>Def.</u> Ein nicht-deterministischer Kellerautomat (PDA) ist ein 6-Tupel

$$M = (Z, \Sigma, \Gamma, \delta, z_0, \#)$$ wobei

- Z endliche Zustandsmenge
- Σ Eingabealphabet

- Γ Kelleralphabet

- $\delta : Z \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to P_e(Z \times \Gamma^*)$ Übergangsfkt.
  (P_e bezeichnet Menge der endl. Teilmengen)

- $z_0 \in Z$ Startzustand

- $\# \in \Gamma$, das untere Kellersymbol.

<u>Idee:</u> $\delta(z, a, A) \ni \underline{(z', B_1 \ldots B_k)}$ bedeutet das:

Wenn M im Zustand z das Zeichen a liest und A oberstes Kellersymbol ist, geht M in Zustand z' und ersetzt A durch die Folge $B_1 \ldots B_k$, wobei B_1 oberstes Symbol im Keller wird.

<u>Bem:</u> $\delta(z, \varepsilon, A) \ni (z', B_1 \ldots B_k)$ ε-Übergang!
Statt Endzustand: Akzeptanz bei leerem Keller.

Def Eine Konfiguration eines Kellerautomaten ist gegeben durch ein Tripel $k \in Z \times \Sigma^* \times \Gamma^*$

Die Relation $\vdash$ auf Konfigurationen ist gegeben durch:

$$(z, a_1 \dots a_n, A_1 \dots A_m) \vdash \begin{cases} (z', a_2 \dots a_n, B_1 \dots B_k A_2 \dots A_m) \\ \qquad \text{falls } \delta(z, a_1, A_1) \ni (z', B_1 \dots B_k) \\ (z', a_1 \dots a_n, B_1 \dots B_k A_2 \dots A_m) \\ \qquad \text{falls } \delta(z, \varepsilon, A_1) \ni (z', B_1 \dots B_k) \end{cases}$$

Sei $\vdash^*$ die reflexive, transitive Hülle. Die durch einen Kellerautomaten M akzeptierte Sprache ist

$$N(M) = \{ x \in \Sigma^* \mid (z_0, x, \#) \vdash^* (z, \varepsilon, \varepsilon) \text{ für ein } z \in Z \}$$

Notation

$$\delta(z, a, A) \ni (z', B_1 \dots B_k)$$

schreiben wir als

$$z\,a\,A \to z'\,B_1 \dots B_k$$

Bsp: $L = \{ a^n b^n \mid n \geq 1 \}$

aabbb

$z_0\ a\ \#\ \to\ z_0\ A$

$z_0\ a\ A\ \to\ z_0\ A\,A$

$z_0\ b\ A\ \to\ z_1\ \varepsilon$

$z_1\ b\ A\ \to\ z_1\ \varepsilon$

$z_0,\ aabbb,\ \#$

$\vdash$

$z_0,\ abbb,\ A$

$\vdash$

$z_0,\ bbb,\ AA$

$\vdash$

$z_1,\ bb,\ A$

$\vdash$

$z_1,\ b,\ \varepsilon$