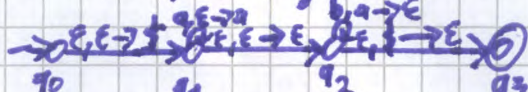
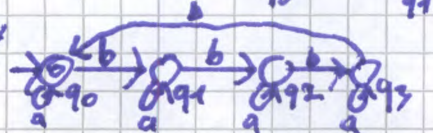
Pumping Lemma L_3 $\{w \in \{a,b\}^* \mid |w|_a > |w|_b\}$ nicht regulär

Betrachte w $a^{n+1}b^n$. Nach PL hat das Wort eine Zerlegung $w=xyz$ mit $|xy| \leq n$. Deshalb kann xy und damit y nur aus einer Folge von a bestehen wobei y nicht das leere Wort sein darf. Also hat die Form $y=a^k$ für ein k mit $1 \leq k \leq n$. Wir pumpen ab und betrachten $xy^0z = xz = a^{n+1-k}b^n \notin L$. ↯ zu PL

Kellerautomat :



b/4



reg. Ausd: $(a|b|c)^*abba(a|b|c)^*$

$G = (V, \Sigma, R, S) \rightarrow (NT, T, P, S)$ $NT = S$ $T = a,b$ $P = S \rightarrow aSbS \mid bSaS \mid \varepsilon$

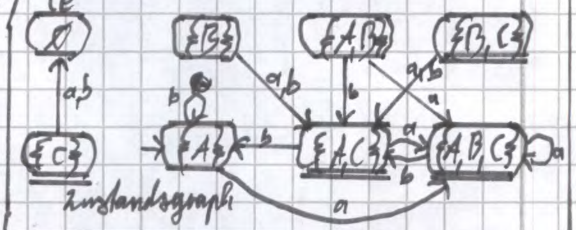
Chomsky: 2NT oder 1T

NEA $N = (\{A,B,C\}, \{a,b\}, \delta, A, \{C\})$

δ:

	a	b	ε
A	A,B	A	∅
B	C	C	C
C	∅	∅	A

R	∅	A	B	C	AB	AC	BC	ABC
E(R)	∅	A	ABC	AC	ABC	AC	ABC	ABC



$L\{0^m0^n \mid n,m \in \mathbb{N}\}$ ist kontextfrei $L\{0^n0^n \mid n \in \mathbb{N}\}$ ist regulär

Eine Zuweisung in einer Programmiersprache besteht aus einem Bezeichner, dem Zuweisungsoperator „=", einem arithm. Ausdruck, der den zuzuweisenden Wert liefert und einem abschl. Semikolon. Arithm. Ausd. besteht aus einer endl. Folge von Operanden, zwischen denen jeweils ein Operator steht. Operanden seien Bezeichner - wie oben, Operationen +, - Alt. arithm. Ausdruck ganze Zahl mit opt. +, -

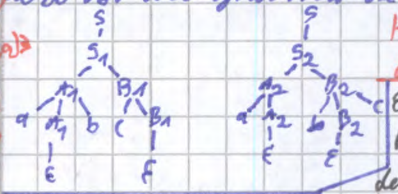
Alphabet: $\Sigma = \{a, b..z, 0, 1...9, =, ;, +, -\}$ | buchstabe ≡ a|b|c...z

ziffer ≡ 0|1|2|...9 id ≡ buchstabe(buchstabe|ziffer)*

op ≡ +|- expression ≡ id(op id)*|(ε|+|-)ziffer$^+$

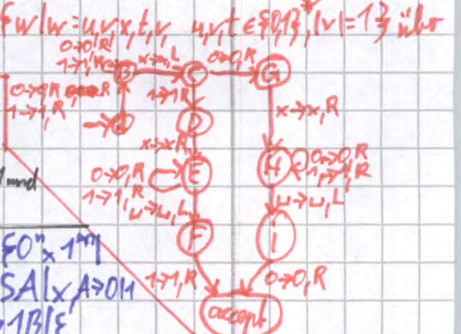
Regulärer Ausdruck einer Zuweisung: buchstabe(buchstabe|ziffer)*=expression;

Sei $G_1 = (\{S, S_1, S_2, A_1, A_2, B_1, B_2\}, \{a,b,c\}, R, S)$ mit $R = \{S \rightarrow S_1 \mid S_2,\ S_1 \rightarrow A_1 B_1,\ A_1 \rightarrow aA_1b \mid \varepsilon,\ B_1 \rightarrow cB_1 \mid \varepsilon,\ S_2 \rightarrow A_2B_2,\ A_2 \rightarrow aA_2 \mid \varepsilon,\ B_2 \rightarrow bB_2c \mid \varepsilon\}$ eine kontextfreie Grammatik. Die von G erzeugte Sprache $L(G_1) = \{a^ib^jc^k \mid i,j,k \in \mathbb{N}_0, (i=j) \vee (j=k)\}$ ist inhärent mehrdeutig. Zeigen sie die Mehrdeutigkeit von G_1 /// Betrachte z.B. das Wort abc. Es hat zwei Ableitungsbäume, also ist die Grammatik G_1 mehrdeutig

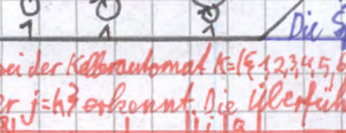


Σ_ε	Γ_ε	$P(Q \times \Gamma_\varepsilon)$
ε	ε	{(2,$)}
a	ε	{(3,ε),(6,a)}
a	ε	{(3,ε)}
b	ε	{(4,b)}
b	ε	{(4,b)}
c	b	{(5,ε)}
c	b	{(5,ε)}
ε	$	{(9,ε)}
a	ε	{(6,a)}
b	a	{(7,ε)}
b	a	{(7,ε)}
c	ε	{(8,ε)}
c	ε	{(8,ε)}
ε	$	{(9,ε)}

Konstruieren sie eine TM T, die die Sprache $L = \{w \mid w = u,v,x,t,v,\ u,v,t \in \{0,1\}^*, |v| = 1\}$ über dem Alphabet $\Sigma = \{0,1,x\}$ erkennt.



Es ist ein deterministischer endlicher Automat E zu konstruieren, der genau diejenigen Wörter über dem Alphabet {0,1} erkennt, in denen mindestens eine 1 und genau zwei 0 auftreten. Spezifizieren sie E als Zustandsgraph.



$\Sigma = \{0,1,x\}$ $L_1 = uxv (u,v \in \{0,1\}^*, |u| = |v|\}$ $L_2 = \{0^n x 1^m \mid m,n \in \mathbb{N}_0\}$ $L(G_1)$: $G_1 = (\{S,A\}, \{0,1,x\}, R_1, S)$ $R_1: S \rightarrow ASA \mid x,\ A \rightarrow 0 \mid 1$

$G_2 = (\{S,A,B\}, \{0,1,x\}, R_2, S)$ $R_2: S \rightarrow AxB,\ A \rightarrow 0A \mid \varepsilon,\ B \rightarrow 1B \mid \varepsilon$

Die Sprache $L_1 \cap L_2 = \{0^n x 1^n \mid n \in \mathbb{N}_0\}$ ist kontextfrei. $G = (\{S\}, \{0,1,x\}, \{S \rightarrow 0S1 \mid x\}, S\}$

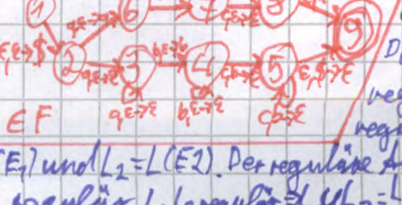
Es sei der Kellerautomat $K = (\{1,2,3,4,5,6,7,8,9\}, \{a,b,c\}, \{a,b,\$\}, \delta, 1, \{9\})$ gegeben, der die kontextfreie Sprache $L = \{a^ib^jc^k \mid i,j,k \geq 1, i=j$ oder $j=k\}$ erkennt. Die Überführungsfunktion δ: *

i	q		
0	1	εabbccε	ε
1	2	abbccε	$
2	3	bbccε	$
3	4	bccε	b$
4	4	ccε	bb$
5	5	cε	b$
6	5	ε	$
7	9		

i	q		
0	1	εabcε	ε
1	2	abcε	$
2	6	bcε	a$
3	7	cε	$
4	8	ε	$
5	9		

abbcc ∈ L, wegen Berechnung und $9 \in F$

abc ∈ L(K) wegen Berechnung und $9 \in F$



$\Sigma = \{a,b,c\}$ $\Sigma^0 \cup \Sigma^2 = \{\varepsilon, aa, ab, ac, ba, bb, bc, ca, cb, cc\}$

Da L_1 und L_2 nach Annahme regulär sind, gibt es reguläre Ausdrücke E_1 und E_2 mit $L_1 = L(E_1)$ und $L_2 = L(E_2)$. Der reguläre Ausdruck $E_1|E_2$ erzeugt nun die Sprache $L_1 \cup L_2$ und von daher ist sie regulär L_1, L_2 regulär $\Rightarrow L_1 \cup L_2 = L(E_1) \cup L(E_2) = L(E_1 + E_2)$ regulär

Wie viele Wörter enthält $\sum_{i=0}^{m} n^i = \frac{n^{m+1}-1}{n-1}$

Gegeben sei die Sprache $L = \{a^i b^j c^k \mid i,j,k \in \mathbb{N}_0, (i=j) \vee (j=k)\}$

$G = (\{S, S_1, S_2, A_1, A_2, B_1, B_2\}, \{a, b, c\}, R, S)$ mit $R = \{S \to S_1 \mid S_2,$

$S_1 \to A_1 \mid B_1,$

$A_1 \to aA_1b \mid \varepsilon,$

$B_1 \to cB_1 \mid \varepsilon,$

$S_2 \to A_2 B_2,$

$A_2 \to aA_2 \mid \varepsilon$

$B_2 \to bB_2c \mid \varepsilon\}$

Ableitung „abc" $S \to S_1 \to A_1 B_1 \to aA_1bcB_1 \to abc$

TM Eine Sprache ist Turing erkennbar falls es eine TM gibt, die L erkennt.
Eine Sprache ist Turing entscheidbar falls es eine Turing-Maschine gibt, die immer hält und L erkennt.