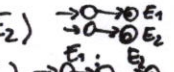
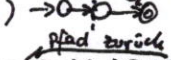
- Alphabet: $\Sigma \neq \emptyset$
- Worte der Länge n: Σ^n
- alle Worte einschl. ε: Σ^* ↳ ist Obermenge aller Sprachen
- $\varepsilon w = w = w\varepsilon$
- $A^0 = \{\varepsilon\}$ mit Sprache A

reguläre Ausdrücke:

- $\emptyset$ ist regEx: $L(\emptyset) = \emptyset$
- a ist regEx: $L(a) = a$
- $E_1|E_2 \Rightarrow L(E_1|E_2) = L(E_1) \cup L(E_2)$ 
- $E_1E_2 \Rightarrow L(E_1E_2) = L(E_1)L(E_2)$ 
- $E_1^* \Rightarrow L(E_1^*) = L(E_1)^*$ 
- $E^+ = EE^*$
- $E^? = \varepsilon|E$

wenn E_1, E_2 reg, dann auch $(E_1|E_2), (E_1E_2), (E_1^*)$

wenn $L_{1/2}$ regulär, dann auch $\overline{L_1}, \overline{L_1^*}, L_1L_2, L_1 \cup L_2, L_1 \cap L_2$

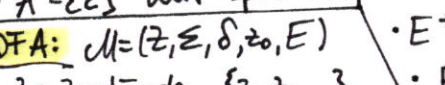
Pumping Lemma (Beweisführung):

- Angenommen L sei regulär
- Nach PL existiert $n > 0$, sodass sich alle $x \in L$ mit $|x| \geq n$ gemäß PL zerlegen lassen
- Sei $x = \ldots$ (passendes suchen)
- x einschränken durch Def PL: 1. $|v| \geq 1$ 2. $|uv| \leq n$ (u, w können ε sein)
- Mit $\exists uv^k w \in L$ für alle $k \geq 0$ zum Widerspruch führen

Wenn es ein $x \in L$ mit $n \leq |x| < 2n$ gibt (n: Anz. Zustände Min-Automat), dann $|L| = \infty$

DFA: $M = (Z, \Sigma, \delta, z_0, E)$

- Z: Zustände $\{z_0, z_1, \ldots\}$
- Σ: Eingabealphabet $\{a, b, \ldots\}$
- δ: Übergangsfkt:



δ	z_0	z_1
0	z_0	z_1
1	z_1	z_0

- z_0: Startzustand
- E: Menge Endzustände $\{z_1, \ldots\}$

Sprache negieren: im DFA Endzustand zu Zustand und alle Zustände zu Endzuständen machen

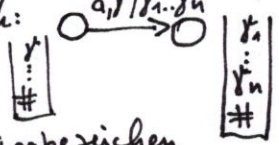
DFA wichtig:
- jeder Knoten hat für jedes Zeichen genau eine Kante
- keine ε-Übergänge

PDA: (kontextfreie Sprache)

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

Γ: Stackalphabet (inkl. #)

#: unterstes Stackzeichen

Stack: 

a: Eingabezeichen

γ vom Stack

$\gamma_1 \ldots \gamma_n$ auf Stack (letztes zuerst)

kontextfreie Grammatik:

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

V: Variablen/Nonterminalsymbole

Σ: Alphabet / Terminalsymbole

P: Regeln

Bsp.: $S \to aS|\varepsilon$
$X^t \to aX^tb$ (in Tiefe wachsend)
$X^t \to XX$ (in Breite wachsend)

Umformungssyntax:
$S \Rightarrow aS \Rightarrow aaS \Rightarrow aa$
bzw. $S \Rightarrow^* aa$ (transitive Hülle)

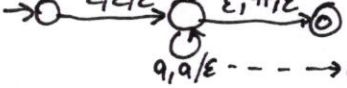
Erweiterte Übergangsfkt:

$\hat{\delta}(z, w) = \begin{cases} z & w = \varepsilon \\ \hat{\delta}(\delta(z,a), x) & w = ax \end{cases}$

Bsp.: $\hat{\delta}(z_0, aaba) = \hat{\delta}(\delta(z_0, a), aba)$
$= \hat{\delta}(z_0, aba) = \ldots = z_E$ ← vorletzter Schritt: $\hat{\delta}(z_E, \varepsilon)$

$\Rightarrow \hat{\delta}$ bestimmt Endzustand des Wortes

Grammatik in PDA:



Ablauf: Stack beginnt mit #
1. |#| 2. |S|#| und dann Startzustand. dann abarbeiten des Wortes

Eindeutigkeit:

Operatorpriorität:
- höhere Priorität weiter „unten"

Assoziativität:
- links-assoziatives nur links

$E \to E+T \mid E-T \mid T$
$T \to T*F \mid T/F \mid F$
$F \to x \mid y \mid z$

NFA: wie DFA, aber kann mehrere Startzustände haben und hat nicht an jedem Knoten für alle Zeichen Kanten

$M = (Z, \Sigma, \delta, S, E)$

CNF (hat keine ε):

Alle Regeln in Form: $A \to BC$ oder $A \to a$ (ist bin. Wurzelbaum bis auf unterste Ebene)

L (ohne ε) in CNF:
1. Terminalsymb. durch neue Variablen ersetzen: $S \to aSa \Rightarrow S \to ASA$, $A \to a$
2. Mehrfache Variablen auf einer Seite ersetzen: $S \to ASA \Rightarrow S \to AR$, $R \to SA$

NFA → DFA:

- Menge von Zustand/Zuständen ist neuer Zustand
- Folgezustand ist Menge von Zuständen, die im NFA mit Zeichen erreicht werden können (ausgehend von der Menge der Ursprungszustände)
- jeder Zustand, in dem Menge ein Endzustand des NFAs ist, ist Endzustand des DFAs

ein DFA/NFA/PDA akzeptiert, wenn Endzustand erreicht wird.

Chomsky Hierarchie:

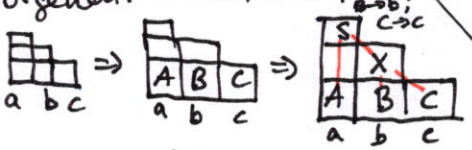
		Regeln: $u \to v$
0.	rekursiv aufzählbar	u, r bel.
1.	kontextsensitiv	$u = \alpha A \beta$, $r = \alpha\gamma\beta$
2.	kontextfrei	$u \in V$, $v \in (V \cup \Sigma)^*$
3.	regulär	$u \in V$, $v \in \Sigma \cup \Sigma V$

Bsp:	Grammatik	Sprache ($n \geq 0$)
0	$ab \to c$	
1	$aAb \to aBb$	$a^n b^n c^n$
2	$A \to aBb$	$a^n b^n$
3	$A \to a$, $A \to aB$	a^n

- Sprache hat mind. gl. Klasse wie Grammatik, kann auch höhere haben
- jede kf enthält reg. Sprache: $\emptyset$
- nicht jede Teilmenge einer reg. Sprache ist kf.: $a^n b^n c^n \subseteq \{a,b,c\}^*$ kontextsensitiv & regulär
- nicht jede Teilmenge einer kf Sprache ist regulär: $a^n b^n \supseteq a^m b^n$ kontextfr. & nicht reg.
- nicht jede Obermenge einer reg. Sprache ist regulär: $a^m b^n \supseteq a^n b^n$ nicht reg.

CYK: braucht CNF!

Vorgehen: $S \to AX$, $X \to BC$, $A \to a$, $B \to b$, $C \to c$



es müssen Laaf alle Regeln gelistet werden:

NFA: möglichst klein!

$\Sigma^n \Sigma^m = \prod_1^n \Sigma \prod_1^m \Sigma = \prod_1^{n+m} \Sigma = \Sigma^{n+m}$

reg. Exp in NFA (möglichst klein): Für jeden Fall NFA aufstellen und dann zusammenführen $\varepsilon|1|1|(011)^*(0|\varepsilon)$ (1. Fall, 2. Fall, 3. Fall, 4. Fall)