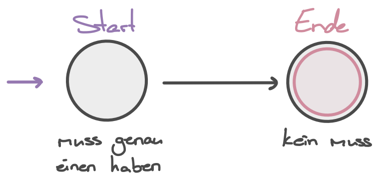
Lukas-Maurice Jakob Sommer 3821176

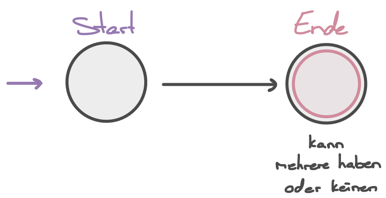
DEA



$A = (X, S, s_0, \delta, F)$

X → Endliches Eingabealphabet *nie leer*
S → Endliche Zustandsmenge *nie leer*
s_0 → Startzustand $s_0 \in S$
δ → Zustandsübergangsfunktion: $\delta: S \times X \to S$
F → Endzustände, $F \subseteq S$

NEA



$A = (X, S, s_0, \delta, F)$

X → Endliches Eingabealphabet
S → Endliche Zustandsmenge
s_0 → Menge der Startzustand $s_0 \subseteq S$
δ → Zustandsübergangsfunktion: $\delta: S \times X \to P(S)$
F → Menge der Endzustände $F \subseteq S$

A = ({Alphabet}, {alle Zustände}, {Startzu.}, δ gemäß Graph, {Endzu.})

der DEA hat keine Mengenklammern

Teilmengenkonstruktion (vom NEA zum DEA)

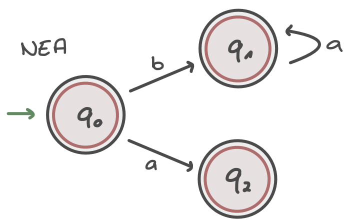
1. alle unerreichbaren Zustände entfernen
2.

δ_1	a	b
q_0	$\{q_2\}$	$\{q_1\}$
q_1	$\{q_1\}$	$\{\}$
q_2	$\{\}$	$\{\}$

beides links hin — wenn $\{q_0, q_1\}$

NEA → DEA

δ_1^d	a	b
$\{q_0\}$	$\{q_2\}$	$\{q_1\}$
$\{q_1\}$	$\{q_1\}$	$\{\}$
$\{q_2\}$	$\{\}$	$\{\}$
$\{\}$	$\{\}$	$\{\}$



NEA: $A = (\{a,b\}, \{q_0, q_1, q_2\}, \{q_0\}, \delta_1$ gemäß Graph, $\{q_0, q_1, q_2\})$

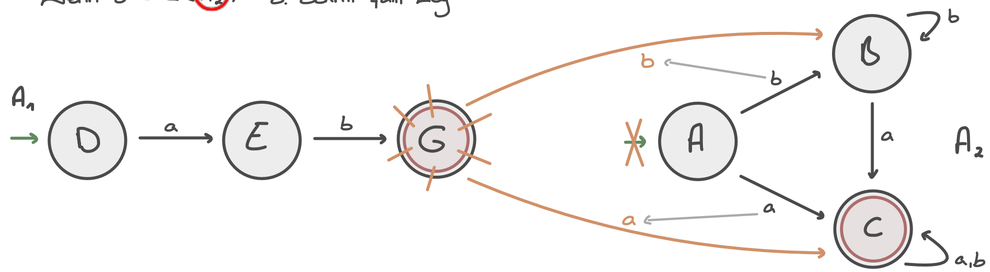
DEA: $A = (\{a,b\}, \{\{q_0\}, \{q_1\}, \{q_2\}, \{\ \}\}, \{q_0\}, \delta_1^d$ gemäß Graph, $\{\{q_0\}, \{q_1\}, \{q_2\}\})$

Doppelklammer

ABGESCHLOSSENHEIT

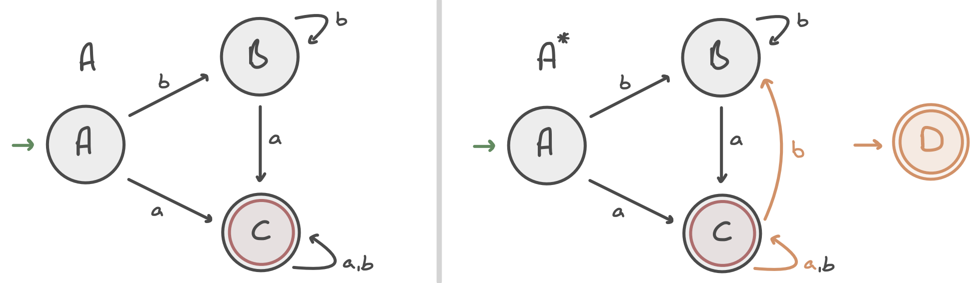
Komplex Produkt $A_1 \cdot A_2$

1. Alle Endzustände von A_1 mit den Nachfolgern der Startzustände von A_2 verbinden
2. Der Automat A_2 verliert alle Startzustandsmarkierungen.
3. Fall $\varepsilon \notin L(A_2)$ (zweite): Der Automat A_1 (Erste) verliert alle Endzustandmarkierungen
   Wenn $\varepsilon \in L(A_2)$ (zweite): 3. Schritt fällt weg



Kleene-Abschluss A^*

1. Alle Endzustände mit den Nachfolgern der Startzustände verbinden
2. Akzeptor für das Leere Wort ergänzen



Komplement $\bar{A}$

- Endzustände werden zu normalen Zuständen
- Normale Zustände werden zu Endzuständen
- nur für DEA's

Vereinigung $A_1 \cup A_2$

- Kasten um beide Automaten
  ↳ Ergebnis ist ein NEA

Schnitt $A_1 \cap A_2$

$$A_1 \cap A_2 = \overline{\overline{A_1 \cap A_2}} = \overline{\bar{A_1} \cup \bar{A_2}}$$

IMMER TUPEL ANGEBEN

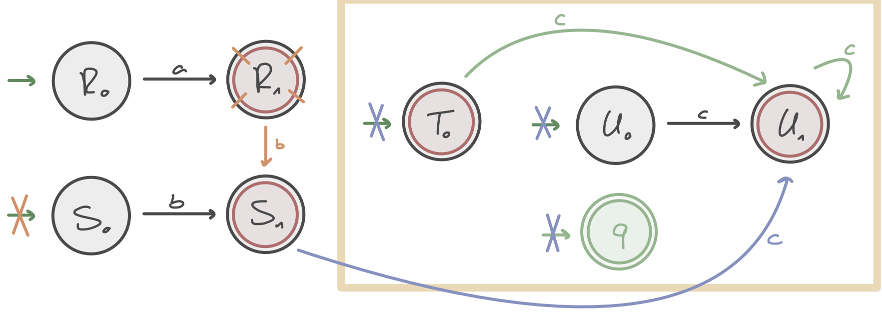
REGULÄRE AUSDRÜCKE

∪ → das oder das

∗ → • → ∪

Konstruktion

$$(a \cdot b) \cdot (\varepsilon \cup c)^*$$



1. Komplexprodukt
2. Vereinigung
3. Kleene Abschluss
4. Komplexprodukt

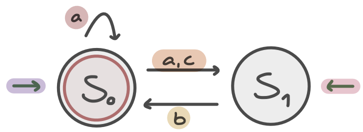
Beispiele

$L = \{ w \in \{a,b,c\}^* \mid w \text{ beginnt mit } a \;\&\; \text{endet mit } b \}$

⇒ $G = a \cdot (a \cup b \cup c)^* \cdot b$

$L = \{ (ab)^n b^m \mid n, m \in \mathbb{N} \}$

⇒ $G = (ab) \cdot (ab)^* \cdot b \cdot b^*$



$R = (a \cup (a \cup c) b)^* \cup b\,(a \cup (a \cup c) b)^*$

GRAMMATIKEN

Typ 0: keine Bedingung, z.B. Linke Seite länger als rechte
allgemeine Grammatik — wenn Startsymbol $S \to \varepsilon$

Typ 1: $|\alpha| \leq |\beta|$, z.B. Links ≤ rechts
nichtverkürzende Grammatik, $aBBa \to abba$, $aBa \to abba$

Typ 2: Links nur Nichtterminale — Strukturbäume (Bsp. nächste Seite)
kontextfreie Grammatik, kontextfrei, $A \to aBC$, $C \to aCe$

Typ 3: Links nur Nichtterminale, rechts nur ein Nichtterminal
(rechts-) lineare Grammatik, regulär, $A \to aA$, $A \to abA$, $A \to a$

$G = (N, T, S, P)$

N → Nichtterminale
T → Terminale
S → Startsymbol
P → Produktion $\alpha \to \beta$

Linkslinear: $aEE \to abE$

rechtslinear: $aEE \to aEb$

Links- & rechtslinear: keine Linearität

Kettenregel: $A \to B$

mehrelementige Terminalketten: $A \to xyzA$
$B \to ab$

Sonderregel Leeres Wort

$S_{neu} \to \varepsilon \mid S_{alt}$ wird erlaubt

S_{neu} wird zum Startzustand

Normalformen

immer eindeutig

Normalformen	Typ-3	Typ-2	Typ-1	Typ-0
$A \to \varepsilon$				×
$A \to t$	×	×	×	×
$A \to tB$	×			×
$A \to BC$		×	×	×
$AB \to CD$			×	×

Backus-Naur-Form

Startsymbol: <S>

P = { <S> ::= a <S> | b <S> | <X>
<X> ::= aaa | bbb }

Normalisierung Typ-2

$G = (\{A, C\}, \{a, b, c\}, A, P)$,
$P = \{ A \to AcA,\ A \to abc,\ A \to C,\ C \to c \}$

1. Seperation von Terminalen

$P' = \{ T_a \to a,\ T_b \to b,\ T_c \to c,$
$A \to AT_cA,\ A \to T_aT_bT_c,\ A \to C,\ C \to c \}$

2. Elimination von Nichtterminalketten länger als 2

$P'' = \{ T_a \to a,\ T_b \to b,\ T_c \to c,$
$A \to AN_1,\ N_1 \to T_cA,$
$A \to T_aN_2,\ N_2 \to T_bT_c,\ A \to C,\ C \to c \}$

3. Elimination von Kettenregeln

$P'' = \{ T_a \to a,\ T_b \to b,\ T_c \to c,$
$A \to AN_1,\ N_1 \to T_cA,$
$A \to T_aN_2,\ N_2 \to T_bT_c,\ A \to c,\ C \to c \}$

Normalisierung Typ-3

$X = \{a, b\}$, $L = \{ (ab)^n b^m \mid n, m \in \mathbb{N} \}$,
$G = (\{S, T\}, \{a, b\}, S, P)$, $P = \{ S \to abS \mid abT,\ T \to bT \mid b \}$

1. Elimination von mehrelementigen Terminalketten

$S \to abS$ — $S \to aA$ — $A \to bS$
$S \to abT$ — $S \to aB$ — $B \to bT$

$P_{NF} = \{ S \to aA,\ A \to bS,\ S \to aB,\ B \to bT,\ T \to bT \mid b \}$

2. Elimination von Kettenregeln — nicht vorhanden in diesem Beispiel

⇒ $G_{NF} = (\{S, T, A, B\}, \{a, b\}, S, P_{NF})$

Automaten erstellen — alle Terminale die alleine stehen bekommen ein N_ε

$P_{NF} = \{ S \to aA,\ A \to bS,\ S \to aB,\ B \to bT,$
$T \to bT \mid b N_\varepsilon,$
$N_\varepsilon \to \varepsilon \}$

Kommas hinsetzen

Beispiel Strukturbaum:

a) $G_1 = (\{A,B,C,D\},$
$\{a,b,c,d\}, A, P_1)$

$P_1 = \{ S \Rightarrow aAd \mid bBd$
$A \rightarrow aAd \mid B$
$B \Rightarrow bBd / C$
$C \rightarrow cC / c$

b) aabcddd

Doppelte Striche!!!

$A \Rightarrow aAd \Rightarrow aaAdd \Rightarrow aaBdd \Rightarrow aabBddd \Rightarrow aabCddd \Rightarrow aabcddd$

Begriffe:

Reguläre Ausdrücke $R = (a \vee b) \cdot (a \vee b)^*$

mengentheoretische Form akzeptierte Sprache: $L(T) = \{a^n b^{2n} \mid n \in \mathbb{N}_0\}$ oder Mengenschreibweise $L = \{x \in \{a,b\}^* \mid x = b^n a^2 b^{3n}$ mit $n \in \mathbb{N}\}$

Konstruiere ... heißt jeweils Schema UND TUPEL

Kontextfreie Grammatik $G = (\{a,b,c\}, \{S,W\}, S, \{S \rightarrow WSb \mid c, W \Rightarrow a \mid b\})$

$G =$ ... Konstuiere den zugeordneten endlichen Automaten $\Rightarrow$ zuerst normalisieren, dann Automat

PUMPING-LEMMA

Sei $L = \{ w w_r \mid w, w_r \in \{0;1\}^* ; w_r$ ist w rückwärts $\}$

immer aufschreiben — passend zur Aufgabe ändern

1. Sei Schranke $p \in \mathbb{N}$
2. Wählen Wort $x \in L$ mit $|x| \geq p$

   $x = 1^p 0 0 1^p \quad ; \quad |x| = 2p \geq p$

3. Zerlege x in $x = uvw$ mit $|uv| / |vw| \leq p$ & $|v| \geq 1$

   $x = 1^p 0 0 1^p = \underbrace{1^l}_{u} \underbrace{1^r}_{v} \underbrace{1^{p-l-r} 0 0 1^p}_{w}$

   mit $l \in \mathbb{N}_0$, $r \in \mathbb{N}$ & $l + r \leq p$ wegen $|uv| \leq p$ &
   $r \geq 1$ wegen $|v| \geq 1$

4. Pumpen von v mit $i \in \mathbb{N}_0$

   $x_i = 1^l (1^r)^i 1^{p-l-r} 0 0 1^p$

   Sei $i = 0$ ⇐ i entweder 0 oder sehr groß setzen KEIN $i = 1$!

   $x_0 = 1^l \cancel{(1^r)^0} 1^{p-l-r} 0 0 1^p = 1^l 1^{p-l-r} 0 0 1^p = 1^{p-r} 0 0 1^p$

   Da $r \geq 1$, ist $1^{p-r} \neq 1^p \Rightarrow x_0 \notin L \Rightarrow$ PL-Komplement gilt
   $p - r \neq p$ — L ist nicht regulär

Beispiele

$L = \{ a^n b^m a^n \mid n, m \in \mathbb{N} \}$

2. $x = a^p b c^p$, $|x| = 2p + 1 \geq p$
3. $x = a^l a^r a^{p-r-l} b a^p$

$L = \{ a^{2n} b^n \mid n \in \mathbb{N} \}$

2. $x = a^{2p} b^p$, $|x| = 2p + p \geq p$
3. $x = a^{2p} b^{p-r-l} b^r b^l$

$L = \{ a^m b^n \mid n, m \in \mathbb{N}, m < n \}$

2. $x = a^p b^{p+1}$, $|x| = p + p + 1 \geq p$
3. $x = a^l a^r a^{p-l-r} b^{p+1}$

$L = \{ a^m b^n \mid n, m \in \mathbb{N}, m > n \}$

2. $x = a^{p+1} b^p$, $|x| = p + 1 + p \geq p$
3. $x = a^{p+1} b^{p-r-l} b^r b^l$

$L = \{ a^m b^n c^i \mid n, m, i \in \mathbb{N}, m + n < i \}$

2. $x = a^p b^1 c^{p+2}$, $|x| = 2p + 3 \geq p$
3. $x = a^l a^r a^{p-l-r} b^1 c^{p+2}$

$L = \{ a^m b^n c^i \mid n, m, i \in \mathbb{N}, m + i < n \}$

2. $x = a^1 b^{p+2} c^p$, $|x| = 2p + 3 \geq p$
3. $x = a^1 b^{p+2} c^{p-l-r} c^l c^r$

$L = \{ x^p y x^k y \mid p, k \in \mathbb{N}, k > p \}$

2. $x = x^s y x^{s+1} y$, $|x| = 2s + 1 \geq s$
3. $x = x^l x^r x^{s-l-r} y x^{s+1} y$

$L = \{ x y^i z^k \mid i, k \in \mathbb{N}, i \geq k \}$

2. $x = x^1 y^p z^p$, $|x| = 1 + 2p \geq p$
3. $x = x^1 y^p z^{p-l-r} z^r z^l$

$L = \{ x^i y^j x^k \mid i \in \mathbb{N}_0 ; j, k \in \mathbb{N}, j < k \}$

2. $x = x^0 y^p x^{p+1}$, $|x| = 2p + 1 \geq p$
3. $x = y^l y^r y^{p-l-r} x^{p+1}$

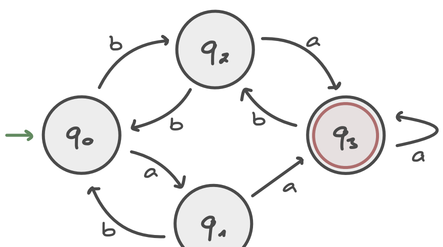
$L = \{ a b^n a u \mid n \in \mathbb{N}, u \in \{c, d\}^*, |u| \leq n + 1 \}$

2. $x = a b^{p+1} a c^p$, $|x| = 2p + 1 \geq p$
3. $x = a b^{p+1} a c^{p-l-r} c^r c^l$

MINIMIERUNG

↳ wenn Pfeile nur vom Automaten weggehen aber nicht hinführen

1. Alle Zustände entfernen die nicht vom Startzustand erreichbar sind
2. Tabelle erstellen & Äquivalenzklassen bilden



1 „normaler"
2 Endzustand
Gruppen definieren

δ	$\sim_0$	$\sim_1$	$\sim_2$	a	$\sim_0$	$\sim_1$	b	$\sim_0$	$\sim_1$
q_0	1	1	1	q_1	1	2	q_2	1	2
q_1	1	2	2	q_3	2	3	q_0	1	1
q_2	1	2	2	q_3	2	3	q_0	1	1
q_3	2	3	3	q_3	2	3	q_2	1	2

Äquivalenzklasse: $\underbrace{\{q_0, q_1, q_2\}}_{1}$ $\underbrace{\{q_3\}}_{2}$

muss alleine bleiben weil Endzustand

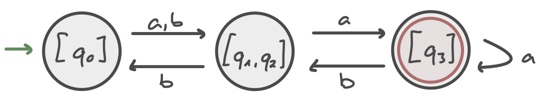
neue Äquivalenzklasse: $\underbrace{\{q_0\}}_{1}$ $\underbrace{\{q_1, q_2\}}_{2}$ $\underbrace{\{q_3\}}_{3}$

Isomorphe Automaten:
- sind bis auf die Benennung der Zustände gleich
- müssen die gleiche Anzahl an Zuständen haben (Endzustände)

wenn nicht äquivalent auch nicht isomorph

Äquivalente Automaten: akzeptieren die selbe Sprache, können verschieden aufgebaut sein

Reduzierter Automat:



$S_R = \{ \{q_0\}, \{q_1, q_2\}, \{q_3\} \}$

$A = (\{a, b\}, S_R, \{q_0\}, \delta$ gem. Graph, $\{ \{q_3\} \})$

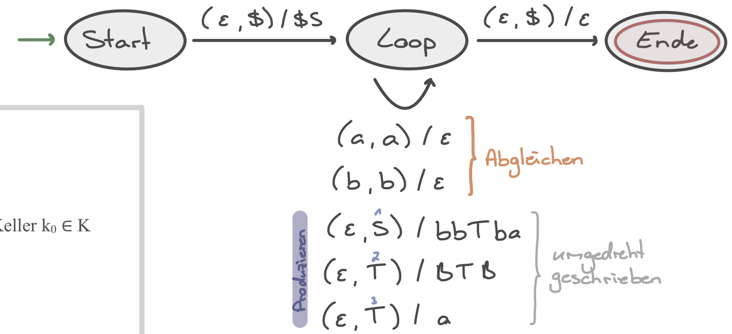
KELLERAUTOMAT

Nichtdeterministischer Kellerautomat

$L = \{ ab^n ab^{n+1} \mid n \in \mathbb{N} \}$, $G = (\{S, T\}, \{a, b\}, S, P)$

$P = \{ S \to ab\overset{1}{T}bb, \; T \to b\overset{2}{T}b \mid \overset{3}{a} \}$

$K_G = (\{a, b\}, \{\$, a, b, S, T\}, \$, \{Start, Loop, Ende\}, Start, \delta$ gem. Graph, $\{Ende\})$



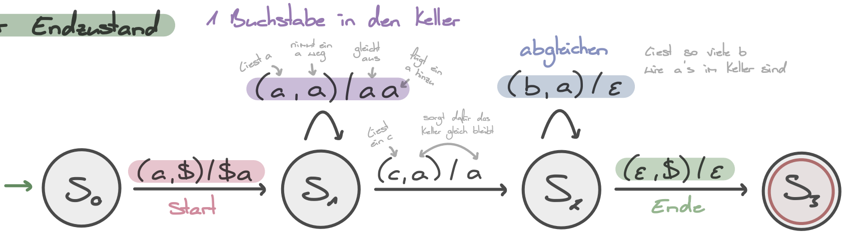
KA = (X, K, k₀, S, s₀, δ, F)

- X → Eingabealphabet
- K → Alphabet der Kellerzeichen
- k_0 → Anfängliches Symbol auf dem Keller $k_0 \in K$
- S → Endliche Menge der Zustände
- s_0 → Startzustand $s_0 \in S$
- δ → Zustandsübergangsfunktion
- F → Menge der Endzustände, $F \subseteq S$

Deterministischer Kellerautomat

Akzeptiert über Endzustand

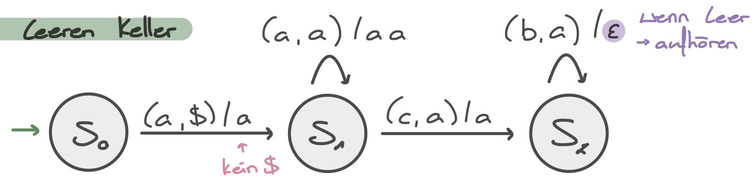
$a^n c b^n$



$KA = (\{a, b, c\}, \underbrace{\{\$, a\}}_{\text{was hinter dem / steht}}, \$, \{S_0, S_1, S_2, S_3\}, S_0, \delta$ gem. Graph, $\{S_3\})$

Akzeptiert über Leeren Keller

$a^n c b^n$



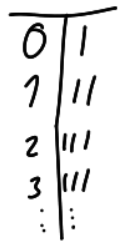
$KA = (\{a, b, c\}, \{\$, a\}, \$, \{S_0, S_1, S_2\}, S_0, \delta$ gem. Graph, $\boxed{}$ $)$

Leerlassen

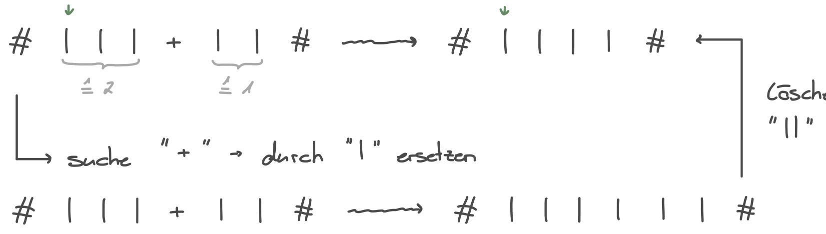
TURINGAUTOMAT

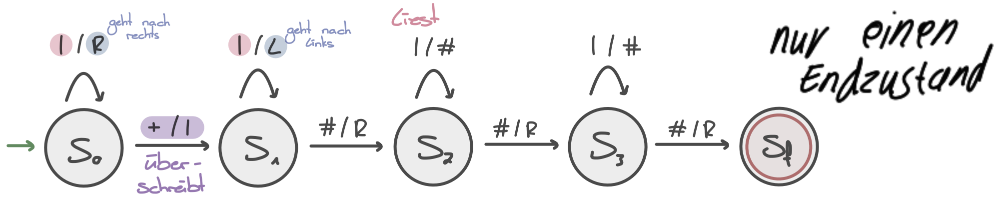
DTA = (X, B, S, s₀, δ, s_f)

- X → Eingabealphabet
- B → Bandalphabet
- S → Endliche Menge der Zustände
- s_0 → Startzustand $s_0 \in S$
- δ → Zustandsübergangsfunktion
- s_f → Haltezustand $s_f \notin S$



Berechnung: 2 + 1





$M = (\{|, +\}, \{|, +, \#\}, \{S_0, S_1, S_2, S_3, \boxed{}\}, S_0, \delta$ gem. Graph, $S_f)$

S_f nicht

Konfigurationsfolge

$(S_0,$	ε	, ||| + ||)	→ $(S_1,$	ε	, |||||)
→ $(S_0,$	|	, || + ||)	→ $(S_1,$	ε	, # |||||)
→ $(S_0,$	||	, | + ||)	→ $(S_2,$	ε	, |||||)
→ $(S_0,$	|||	, + ||)	→ $(S_2,$	ε	, # ||||)
→ $(S_1,$	|||	, | ||)	→ $(S_2,$	ε	, ||||)
→ $(S_1,$	||	, | | ||)	→ $(S_2,$	ε	, # ||||)
→ $(S_1,$	|	, ||||)	→ $(S_f,$	ε	, ||||)

Aufgaben: Sind folgende Aussagen richtig?

a) $(L_1 \cup L_2)L_3 = L_1L_3 \cup L_2L_3$ ✓

Seien $L_1=\{a\}, L_2=\{aa\}, L_3=\{\varepsilon\}$

$(\{a\} \cap \{aa\}) \cdot \{\varepsilon, a\} = \{\} \cdot \{\varepsilon, a\} = \{\}$

b) $(L_1 \cap L_2)L_3 = L_1L_3 \cap L_2L_3$ Nein: $\{a\} \cdot \{\varepsilon, a\} \cap \{aa\} \cdot \{\varepsilon, a\} = \{a, aa\} \cap \{aa, aaa\} = \{aa\}$

c) $(L_1 \cup L_2)^* = L_1^* \cup L_2^*$ Nein: $ab \in (\{a\} \cup \{b\})^*$; $ab \notin \{a\}^* \cup \{b\}^*$

d) $(L_1 \cap L_2)^* = L_1^* \cap L_2^*$ Nein: $(\{a\} \cap \{aa\})^* = (\emptyset)^* = \{\varepsilon\} \neq \{\varepsilon, aa\}^* = \{a\}^* \cap \{aa\}^*$

e) $(L_1^*)^* = L_1^*$ ✓

f) $(L_1^+)^+ = L_1^+$ ✓

g) $(L_1L_2)^*L_1 = L_1(L_2L_1)^*$ ✓

h) $(L_1L_2)^+L_1 = L_1(L_2L_1)^+$ ✓

$a^n \rightarrow$ Typ 3

$a^nb^n \rightarrow$ Typ 2

$a^nb^nc^n \rightarrow$ Typ 1

$a^nb^nc^nd^n \rightarrow$ auch Typ 1