Zusammenfassung

Grundbegriffe

Die Mengen *aller* Teilmengen von Q heißt **Potenzmenge** von Q, kurz $\wp(\mathrm{Q})$
Die Potenzmenge von $M = \{a, b, c\}$ ist
$\wp(\mathrm{M}) = \{\emptyset, \{\mathrm{a}\}, \{\mathrm{b}\}, \{\mathrm{c}\}, \{\mathrm{a}, \mathrm{b}\}, \{\mathrm{a}, \mathrm{c}\}, \{\mathrm{b}, \mathrm{c}\}, \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}\}$

$A = \{a, b, c\}$
$A^* = \{\underbrace{\varepsilon,}_{A^0} \underbrace{a, b, c,}_{A^1} \underbrace{aa, ab, ac, ba, bb, bc, ca, cb, cc,}_{A^2} aaa, \dots\}$
$A^+ = \{a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, \dots\}$

Die gesamte Wortmenge über A setzt sich dann aus den Wörtern dieser unendlich vielen disjunkten Teilmengen A^i zusammen:

$$A^* = \coprod_{i \in \mathbb{N}} A^i \qquad A^+ = A^* \backslash \{\varepsilon\}$$

Definition 1.5:

Zwei Mengen M_1 und M_2 heißen *gleichmächtig*, wenn es eine *bijektive* Abbildung von M_1 auf M_2 gibt.

Ist eine der **beiden Mengen die der natürlichen Zahlen $\mathbb{N}$, so ist M eine abzählbar unendliche Menge**, kurz: „M ist abzählbar unendlich".

Speziell für $M = \mathbb{N}$ ist offensichtlich, dass die Menge der natürlichen Zahlen $\mathbb{N}$ eine **abzählbar unendliche Menge** ist. Im Gegensatz dazu ist die Menge der reellen Zahlen $\mathbb{R}$ *überabzählbar* unendlich. Es gibt also (sehr viel) mehr reelle Zahlen als natürliche.

Sprache
Sei A ein Alphabet. Jede Teilmenge $L \subseteq A^*$ heißt **Sprache** über A.
Gehört ein Wort $w \in A^*$ zur Sprache $L \subseteq A^*$?

Sprachen sind Wortmengen und besitzen höchstens abzählbar unendliche viele Elemente.

L sei die Sprache über $A = \{\mathrm{a}\}$, die aus einer primzahligen Anzahl von a's bestehen. Geben Sie eine mathematische Notation dafür an.

$L = \{a^p \mid p \text{ ist Primzahl}\},\ L = \{w \mid w \text{ hat die Form } a(a,b)^*b \text{ oder } b(a,b)^*b(a,b)^*b\}$

Parsing: Für jedes Wort aus der zugehörigen Wortmenge T* muss entschieden werden können, ob entweder $w \in L(G)$ oder $w \notin L(G)$ gilt. In der Informatik (Compilerbau) spricht man von **Parsing**.

Beispiel 4.4:

Die Potenzmenge von $M = \{a, b, c\}$ ist

$\wp(M) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$.

Für eine endliche Menge M besitzt $\wp(M)$ genau $2^{|M|}$ Elemente. In Beispiel 4.4 sind das gerade $2^{|M|} = 2^3 = 8$.

Formale Grammatik

Zur Definition einer natürlichen Sprachen benötigen wir eine formale Grammatik.

Definition 2.1:

Eine (formale) **Grammatik** G ist ein 4-Tupel $G = (N, T, P, s)$ mit folgenden Eigenschaften:

N ... Menge der **Nichtterminale** (grammatikalische Variablen)

T ... Menge der **Terminale** (Alphabetzeichen)
N, T sind nichtleere, endliche und disjunkte Mengen, d.h. $N \cap T = \varnothing$.

P ... endliche Menge von **Regeln** oder **Produktionen**

$$P = \{\alpha \to \beta | \alpha \in (N \cup T)^* \setminus T^* \text{ und } \beta \in (N \cup T)^*\}$$

s ... **Start**- oder **Satz**- oder **Spitzensymbol**, mit $s \in N$.

Definition 2.4:

Die durch G definierte (oder auch erzeugte oder erzeugbare) Sprache $L(G)$ ist
$L(G) = \left\{ w \mid w \in T^* \text{ und } s \Rightarrow_G^* w \right\}$.

Definition 2.7:

Zwei Grammatiken G_1 und G_2 heißen *äquivalent*, geschrieben: $G_1 \equiv G_2$, wenn die zugehörigen erzeugbaren Sprachen übereinstimmen, d.h., wenn $L(G_1) = L(G_2)$.

Chomsky-Hierarchie

Typ	Bezeichnung	Regelgestalt	Beispiele
0	unbeschränkt	keine Einschränkung	$\alpha A \beta B \to \alpha\alpha\gamma\beta\beta$
1	kontextsensitiv	wie Typ 0 und zusätzlich: *längenmonotone* Regeln, d. h. $\lvert\alpha\rvert \le \lvert\beta\rvert$; Ausnahme: $s \to \varepsilon$ darf vorkommen, wenn s in keiner Regel auf der rechten Seite steht	$\lvert w_1\rvert \le \lvert w_2\rvert$ $\alpha A \beta \to \alpha\gamma\beta$ $S \to \varepsilon$
2	kontextfrei	wie Typ 1 und zusätzlich: $\alpha \in N$; Ausnahme: Regeln der Form $\alpha \to \varepsilon$ sind erlaubt	$\lvert w_1\rvert = 1$ $A \to aABa$
3	regulär	wie Typ 2 und zusätzlich: Neben Regeln der Form $\alpha \to x$ gibt es entweder *nur* linkslineare, d. h. $\alpha \to Ax$, oder *nur* rechtslineare, d. h. $\alpha \to xA$, mit $x \in T$ und $A \in N$	$A \to aB$ *oder* $A \to Ba$

Kontextsensitive Sprache $L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}, n \ge 1\}$

Kontextfreie Sprache $L(G) = \{a^n b^n \mid n \in \mathbb{N}, n \ge 1\}$

Reguläre Sprache $L(G) = \{a^n \mid n \in \mathbb{N}, n \ge 1\}$

$\mathscr{L}_{\text{LBTM}} = \mathscr{L}_{\text{Typ1}}$

$\mathscr{L}_{\text{kfS}} = \mathscr{L}_{\text{NKA}}$ $\quad \mathscr{L}_{DKA} \subseteq \mathscr{L}_{NKA}$

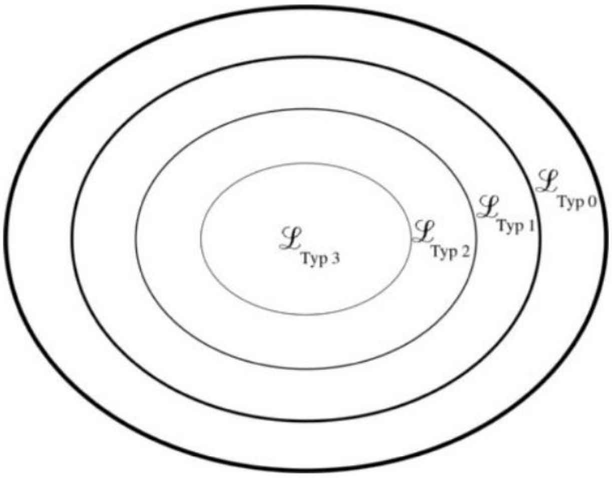
$\mathscr{L}_{NEA_\varepsilon} \equiv \mathscr{L}_{DEA} \equiv \mathscr{L}_{rG}$

Reguläre Sprache $L(G) = \{w \mid w \in \{a, b\}^* \text{ beginnt oder endet mit } a\}$
Regulärer Ausdruck = $a(a + b)^* + (a + b)^* a$

oder

Zusammenfassung

Die von Chomsky definierten Sprachklassen bilden eine Hierarchie:



$\mathcal{L}_{endlich} \subset \mathcal{L}_{Typ3} \subset \mathcal{L}_{Typ2} \subset \mathcal{L}_{Typ1} \subset \mathcal{L}_{Typ0}$

ε-Sonderregelungen

Die ε-Regeln verstoßen gegen die **Längenmonotonie**. $Bsp.\ A \rightarrow \varepsilon$
$Regel: |\alpha| \leq |\beta|\ Aber\ |\alpha| > 0\ \&\ |\beta| = |\varepsilon| = 0\ Somit\ |\alpha| > |\beta|$
Die Verwendung solcher ε-Regeln macht die Formulierung einer Grammatik bequemer.

> Zu jeder ε-freien $Grammatik_{Typ1,2,3}$ gibt es eine äquivalente $Grammatik'_{Typ1,2,3}$
> $L(G') = L(G) \cup \{\varepsilon\}$

Grammatiktransformation für **ksG** und **kfG**:

a) Wähle ein noch nicht in N vorhandenes Nichtterminal s als Spitzensymbol von G.
b) Ergänze die Regeln $S' \rightarrow s\ und\ s' \rightarrow \varepsilon$
   Aus G entsteht die Grammatik
   $G' = (N \cup \{s'\}, T, P \cup \{s' \rightarrow s|\varepsilon\}, s')$

Bsp. $G = (\{K,H\},\{i,j,l,r\},\{K \rightarrow iHj, H \rightarrow lK|r\},K)$
$G' = (\{K,H,\boldsymbol{Q}\},\{i,j,l,r\},\{\underbrace{\boldsymbol{Q}}_{a)} \rightarrow \underbrace{\boldsymbol{K}|\boldsymbol{\varepsilon}}_{b)}, K \rightarrow iHj, H \rightarrow lk|r\},\boldsymbol{Q})$

Grammatiktransformation für **reguläre Grammatiken**:

a) Wähle ein noch nicht in N vorhandenes Nichtterminal s‘ als Spitzensymbol von G‘.
b) Ergänze für alle Regeln $s \rightarrow$ ß in P die Regeln $s' \rightarrow$ ß sowie $s' \rightarrow \varepsilon$.

Bsp. $G = (\{S,H\},\{a,b\},\{S \rightarrow aS \mid aH, H \rightarrow b\},S)$
$G' = (\{S,H,\boldsymbol{Q}\},\{a,b\},\{\underbrace{\boldsymbol{Q}}_{a)} \rightarrow \underbrace{aS \mid aH \mid \varepsilon}_{b)}, S \rightarrow aS| aH, H \rightarrow b\},Q)$

ε-Freiheit herstellen

Zu jeder $Grammatik_{Typ1,2,3}$ gibt es eine äquivalente $Grammatik'_{Typ1,2,3}$ ohne ε-Regeln. (ggf. bis auf $S' \to \varepsilon$).

a) Zunächst werden alle Nichtterminale gesucht, die zu ε abgeleitet werden können, ohne dass ein neuer Nichtterminal entsteht. $N_\varepsilon = \{A_i\}$ für alle $A_i \to \varepsilon$
b) Entferne ε und füge in jeder Regel das Nichtterminal hinzu.

Bsp. $G = (\{S, B, T\}, \{a, b\}, P, S)$
$P = \{S \to abS \mid aS \mid aB,$
$B \to aB \mid T,$
$T \to abT \mid \boldsymbol{\varepsilon}\}$
$N_\varepsilon = \{B, T\}$

$P' = \{S \to abS \mid aS \mid aB \mid a,$
$B \to aB \mid T \mid a,$
$T \to abT \mid ab\}$

a) Mit B, T kann zu ε Abgeleitet werden.

b) Nun wird in jeder Regel, wo ein **Nichtterminal + Terminal** aus a) steht, eine neue Regel mit dem Nichtterminal hinzugefügt. ε wird entfernt.

Das Wortproblem

Das Wortproblem ist für Typ-1,2,3 Sprachen entscheidbar, jedoch nicht für Sprachen vom Typ 0.

Wenn die betrachtete Satzform β eine Länge besitzt, die größer als $n = |w|$ ist, wird diese Satzform nicht in S_{i+1} übernommen. Aufgrund der für ksG geforderten **Längenmonotonie** wäre sie eine (für die Ableitung von w) aussichtslose Kandidatin.

Bsp. $G = (\{A, B\}, \{a, b, c\}, \{A \to aABc \mid aBc, cb \to Bc, aB \to ab, bB \to bb\}, A)$
$\{S_0\} = \{A\}$
$\{S_1\} = \{A, aABc, aBc\}$
$\{S_2\} = \{A, aABc, aBc, \cancel{aaABcBc}, aaBcBc, \cancel{abc}\}$, da $|aaABcBc| = 7 > |w|$ und $abc \neq w$
$\{S_3\} = \{A, aABc, aBc, aaBcBc, aabcBc, aaBBcc\}$
$\{S_4\} = \{A, aABc, aBc, aaBcBc, aabcBc, aaBBcc, aabBcc$
$\{S_5\} = \{A, aAbc, aBc, aaBcBc, aabcBc, aaBBcc, aabBcc, aabbcc\}$ und Stopp
da $aabbcc = w$

Beweis:

Die Längenmonotonie der Grammatik ist die Voraussetzung dafür, dass das oben beschriebene Verfahren im Allgemeinen terminiert. Chomsky-Typ-0-Grammatiken müssen nicht längenmonoton sein.

Mehrdeutigkeit kontextfreier Grammatiken

Eine KfG ist *mehrdeutig*, wenn es ein Wort in $L(G)$ gibt, das (mindestens) zwei Linksableitungen besitzt. Andernfalls ist die Grammatik eindeutig.
Eine kfS ist eindeutig, wenn es (mindestens) ein eindeutige kfG G, mit $L = L(G)$, gibt. Ansonsten ist L (inhärent) mehrdeutig.

Bsp. Für das Wort *abc* gibt es zwei Linksableitungen: $S \to aB \to abc$ und $S \to Ac \to abc$.

Äquivalenz

Zwei Grammatiken G_1 und G_2 heißen *äquivalent*, geschrieben $G_1 \cong G_2$, wenn die zugehörigen erzeugbaren Sprachen übereinstimmen, d.h., wenn $L(G_1) = L(G_2)$.

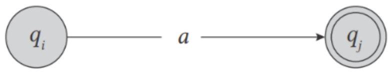
Reguläre Sprachen

Deterministischer endlicher Automat (DEA, EA)

Endliche Automaten definieren reguläre Sprachen.

Ein **deterministischer endlicher Automat**, kurz: DEA (auch EA) ist ein Quintupel $M = (Q, \Sigma, \delta, q_0, \mathrm{E})$ welche **reguläre Sprachen definiert.**

Q … endliche Menge der Zustände
Σ … Eingabealphabet, $Q \cap \Sigma = \emptyset$
δ … Überführungsfunktion (totale Funktion), $Q \times \Sigma \to \mathrm{Q}$
q_0 … Startzustand, $q_0 \in Q$
E … endliche (nichtleere) Menge der Endzustände, $E \subseteq Q$



$\delta(q_i, a)$ $\quad q_i \in Q \text{ und } a \in \Sigma$, definiert den Folgezustand
$\hat{\delta}(q, \varepsilon) = q$ ←Der Automat drückt aus, dass er für $w = \varepsilon$ in Zustand q stoppt
$\hat{\delta}(q, aw) = \hat{\delta}(\delta(q, a), w)$ Nach dem Zustandswechsel von q zu $\delta(q, a)$ wird mit dem Restwort w fortgesetzt.

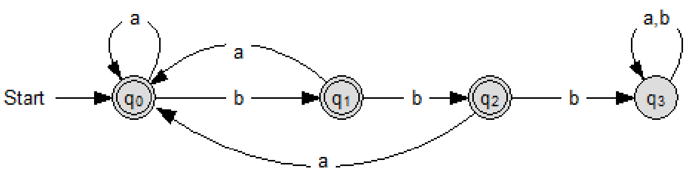
Definition 4.4:

Unter Verwendung von $\hat{\delta}$ kann die von einem DEA akzeptierte Sprache wie folgt beschrieben werden:

$$L(M) = \left\{ w \mid w \in \Sigma^* \text{ und } \hat{\delta}(q_0, w) = q_e \text{ und } q_e \in E \right\}$$

Beispiel deterministischer endlicher Automat (DEA)

DEA $M = (Q, \Sigma, \delta, q_0, \mathrm{E}) = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, \mathrm{q}_0, \{q_0, q_1, q_2\})$

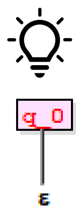


δ	a	b
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_0	q_3
q_3	q_3	q_3

$L(M)$ *ist die Menge aller Wörter über* $\{a, b\}^*$, *die höchstens zwei* b'*sam Stück enthalten.* ($D.h.$ *genau* 0,1 *oder* $2b's$)

$G = (N, T, P, s) = (\{q_0, q_1, q_2\}, \{a, b\}, P, q_0)$
$P = \{q_0 \to a \mid b \mid b\, q_1 \mid a\, q_0 \mid \varepsilon\,,$
$\quad q_1 \to a\, q_0 \mid a \mid b\, q_2 \mid b\,,$
$\quad q_2 \to a \mid a\, q_0 \,\}$

Auch das leere Wort gehört zur Wortmenge dazu

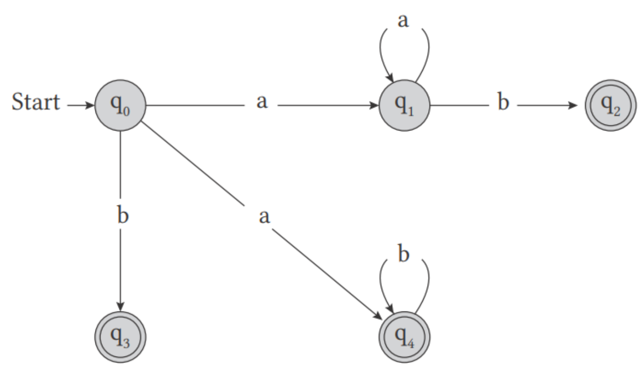


Nichtdeterministischer endlicher Automat (NEA)

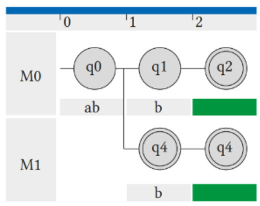
$M = (Q, \Sigma, \delta, q_0, \mathrm{E})$
Q … endliche Menge der Zustände
Σ … Eingabealphabet, $Q \cap \Sigma = \emptyset$
δ … Überführungsfunktion (totale Funktion), $Q \times \Sigma \to \wp(Q)$
q_0 … Startzustand, $q_0 \in Q$
E … endliche (nichtleere) Menge der Endzustände, $E \subseteq Q$

Beispiel 4.5:

Der NEA $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{a,b\}, \delta, q_0, \{q_2, q_3, q_4\})$ mit



δ	a	b
q_0	$\{q_1, q_4\}$	$\{q_3\}$
q_1	$\{q_1\}$	$\{q_2\}$
q_2	$\{\ \}$	$\{\ \}$
q_3	$\{\ \}$	$\{\ \}$
q_4	$\{\ \}$	$\{q_4\}$



akzeptiert die Sprache $L = \{w \mid w = a^n b \text{ oder } w = ab^n, n \geq 0\}$.

Akzeptiert mehr als ein Automat das Eingabewort, so ist die Grammatik mehrdeutig

Linkslineare Grammatik in rechtslineare transformieren

a) Transformation aller Regeln in linkslineare Regeln: $X \to Ya$ bleiben unverändert. $X \to a$ werden ersetzt durch $X \to Ha$ und $H \to \varepsilon$.

b) Transformation der vorbereiteten Regeln in rechtslineare: Wähle H als neues Spitzensymbol und streiche die Regel $H \to \varepsilon$. Ergänze $s \to \varepsilon$, wobei s das Spitzensymbol der linkslinearen Grammatik ist. Ersetzte jede Regel der Form $X \to Ya$ durch $Y \to aX$, Falls $\varepsilon \in L(G)$, muss $H \to \varepsilon$ wieder hinzugefügt werden.

Bsp. $G = (\{A, B, S\}, \{a, b, c\}, P, S)$ $P = \{S \to Bc \mid Ac,\ A \to a \mid Aa, B \to b \mid Bb\}$

a) $G' = (\{A, B, S, H\}, \{a, b, c\}, P', S)$
$P' = \{S \to Bc \mid Ac;\ A \to Ha \mid Aa; B \to Hb \mid Bb; H \to \varepsilon\}$

b) $G'' = \{A, B, S, H\}, \{a, b, c\}, P'', H)$
$P' = \{S \to Bc \mid Ac;\ A \to Ha \mid Aa; B \to Hb \mid Bb; \cancel{H \to \varepsilon}\}$
$P'' = \{H \to aA \mid bB; A \to cS \mid aA; B \to cS \mid bB; S \to \varepsilon\}$

$\boxed{\begin{array}{l} X \to Ya \\ Y \to aX \end{array}}$

Konstruktion eines NEA aus einer regulären Grammatik

1. ε-Freiheit herstellen (siehe oben)
2. Linkslineare Regeln in rechtslineare Regeln transformieren (siehe letzten Schritt)
3. Allgemeine Konstruktionsprinzip:
   Gegeben: reguläre Grammatik $G = (N, T, P, s)$ ohne ε-Regeln
   Gesucht: NEA $M = (Q, \Sigma, \delta, q_0, \mathrm{E})$, mit $L(M) = L(G)$
   $\Sigma = T$ und $q_0 = s$
   $Q = N \cup \{q_x\}$, mit $q_x \notin N$, und δ wie folgt:
   Falls $A \rightarrow aB$ in P existiert, so füge $\delta(\mathrm{A}, \mathrm{a}) \ni \mathrm{B}$ in δ hinzu
   Falls $B \rightarrow a$ in P existiert, so füge $\delta(\mathrm{B}, \mathrm{a}) \ni \mathrm{q}_x$ in δ hinzu
   Für die Menge der Endzustände E des NEA M gilt

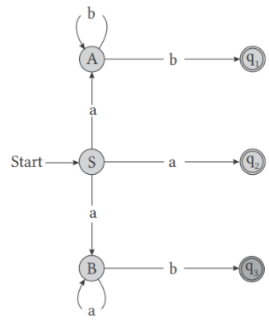
δ	a	...
A	B	...
B	q_x	...

q_x sind Endzustände, welche hinzugefügt werden.

$$E = \begin{cases} \{q_0, q_x\}, & \text{wenn} \quad (s \rightarrow \varepsilon) \in P \\ \{q_x\}, & \text{sonst} \end{cases}$$

Bsp. $G = (\{S, A, B\}, \{a, b\}, \{S \rightarrow aA \mid aB \mid a, A \rightarrow b \mid bA, B \rightarrow b \mid aB\}, S)$
$M = (Q, \Sigma, \delta, q_0, \mathrm{E})$
$\Sigma = T = \{a, b\}$
$q_0 = s = S$
$E = \{q_1, q_2, q_3\}$
$Q = \{S, A, B, q_1, q_2, q_3\}$

δ	a	b
S	$\{A, B, q_2\}$	$\{\}$
A	q_0	$\{q_1, A\}$
B	B	q_3
q_1	$\{\}$	$\{\}$
q_2	$\{\}$	$\{\}$
q_3	$\{\}$	$\{\}$



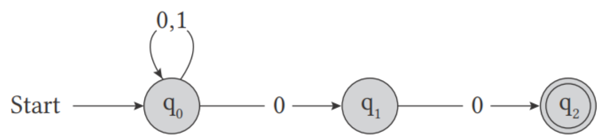
NEA zu einem äquivalenten DEA konstruieren

Jedes Element (also jeder Menge) aus $\wp(Q)$ wird genau ein Zustand z_i zuzuordnen. Dann kann die Überführungsfunktion δ' des gesuchten DEA M' aus δ des betrachteten NEA $M = (Q, \Sigma, \delta, q_0, E)$ nach Vorschrift
$\delta'(z, a) = \coprod_{q \in z} \delta(q, a)$ gebildet werden. Die anderen Bestandteile von $M' = (Q', \Sigma', \delta', z_0, E')$ ergeben sich aus

$Q' = \wp(Q)$ nach Umbenennung zu z_i, mit $0 \leq i \leq 2^{|Q|}$
$z'_0 = z_0 = \{q_0\}$
$E' = \{R | R \subseteq Q \text{ und } R \cap E \neq \emptyset\}$ nach Umbenennung zu z_i

Beispiel

Konstruktion eines äquivalenten DEA $M' = (Q', \Sigma', \delta', z_0, E')$ aus einem NEA $M = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_2\})$ mit



δ	0	1
q_0	$\{q_0, q_1\}$	q_0
q_1	q_2	$\{\ \}$
q_2	$\{\ \}$	$\{\ \}$

$$\wp(Q) = \left\{ \underbrace{\{q_0\}}_{z_0}, \underbrace{\{q_1\}}_{z_1}, \underbrace{\{q_2\}}_{z_2}, \underbrace{\{q_0, q_1\}}_{z_3}, \underbrace{\{q_0, q_2\}}_{z_4}, \underbrace{\{q_1, q_2\}}_{z_5}, \underbrace{\{q_0, q_1, q_2\}}_{z_6}, \underbrace{\varnothing}_{z_7} \right\}$$

$$\begin{aligned} \delta'(z_3, 0) &= \delta'(\{q_0, q_1\}, 0) = \delta(q_0, 0) \cup \delta(q_1, 0) \\ &= \{q_0, q_1\} \cup \{q_2\} = \{q_0, q_1, q_2\} \\ &= z_6 \end{aligned}$$

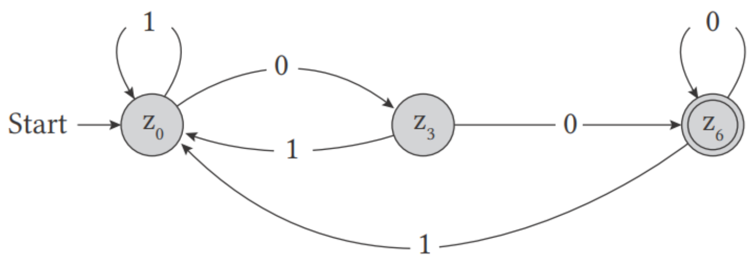
δ	0	1
q_0	$\{q_0, q_1\}$	q_0
q_1	q_2	$\emptyset$
q_2	$\emptyset$	$\emptyset$

δ′	0	1	δ′
q_0	$\{q_0, q_1\}$	q_0	z_0
$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	q_0	z_3
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	q_0	z_6

Wenn sich hier ein Endzustand befindet, wird der neue auch Endzustand

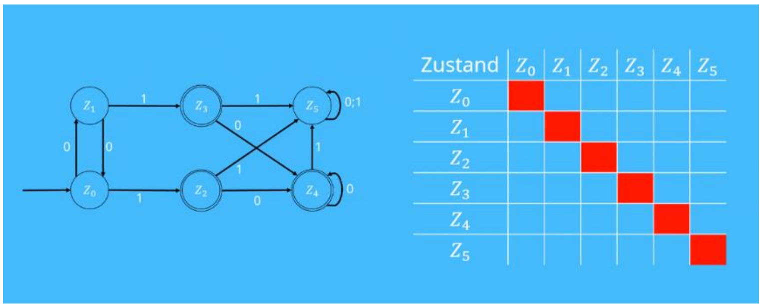
(Der Folgezustand von)

$\delta'(z_0, 0) = \delta'(\{q_0\}, 0) = \delta(q_0, 0) = \{q_0, q_1\} = z_3$

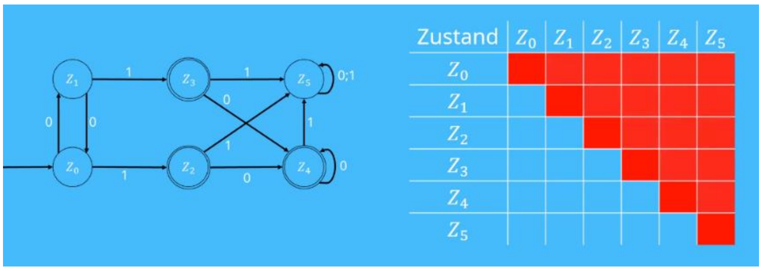


Minimalautomat

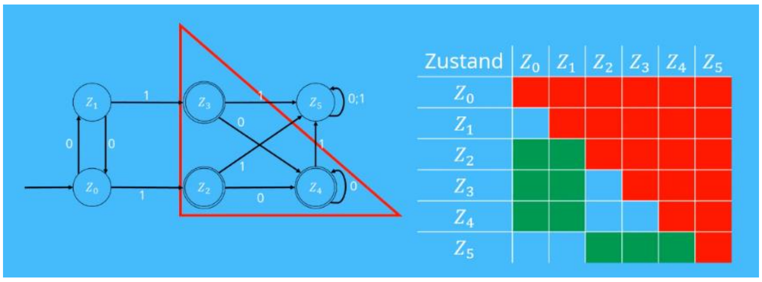
Da Zustände nicht mit sich selbst überprüft werden müssen, können wir die **Diagonale der Tabelle** komplett **streichen**.



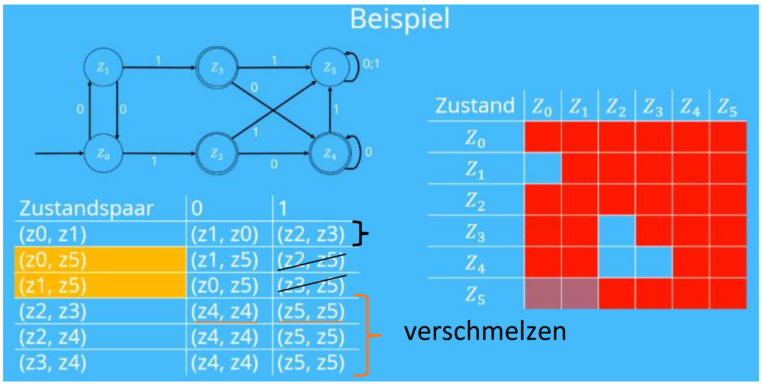
Außerdem werden die **Paare nur in eine Richtung betrachtet**, wir müssen die Paare (z1, z2) und (z2, z1) also nicht als einzelne Paare sehen und können damit die **obere Hälfte der Tabelle über der Diagonalen** (in der Abbildung rot markiert) **streichen**.



Markiere nun alle Paare, wenn genau **einer der beiden Zustände**, also entweder z oder z, ein Endzustand ist.



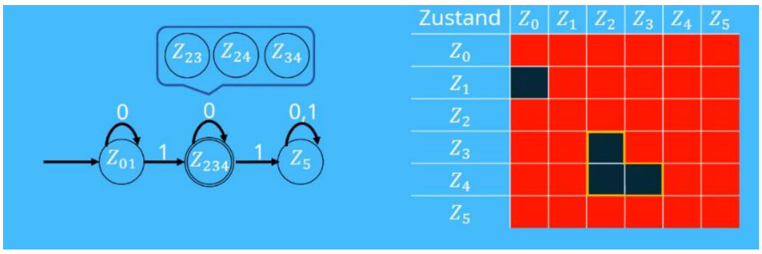
Wir testen jetzt für jedes Zustandspaar die Folgezustände bei der Eingabe der Zeichen 0 und 1. In der ersten Zeile wird Beispielsweise der Zustand z0 bei Eingabe einer 0 zu z1, während der Zustand z1 zu z0 wird.



Wir suchen uns nun die Zustandspaare, die bei einer Eingabe ein Zustandspaar ergeben, das bereits gestrichen ist. Ein Beispiel ist das Zustandspaar (z0, z5)->z2,z5. Finden wir **ein** paar, so wird das Zustandspaar gestrichen.

> Wenn beim Überprüfen eines Zustandspaares ein Zustand (z_i, z_i) entsteht, wird dieser nicht gestrichen.

Die **übrig gebliebenen Paare** können wir nun **kombinieren**.



Reguläre Ausdrücke

Die Idee besteht darin, zu deren Definition nur ganz wenige Basismengen und insgesamt nur **drei Mengenoperationen** zu verwenden. Durch deren wiederholte und kombinierte Anwendung kann jede beliebige reguläre Menge (Sprache) erzeugt werden. Die **einzigen Basismengen** sind:

$\emptyset$	Die leere Menge, die also kein einziges Wort enthält, nicht einmal das leere Wort.
$\{\varepsilon\}$	Die Menge, die nur das leere Wort ε enthält.
$\{a\}$	Alle Einermengen, die nur jeweils genau ein ein-zeichiges Wort, das aus dem zugehörigen Alphabet Zeichen besteht, enthalten.

Die so definierten Basismengen sind reguläre Mengen. Durch Anwendung der im Folgenden angegebenen Operationen – und **nur durch diese** – entstehen weitere reguläre Mengen.

∪	Vereinigung: $A \cup B = \{w \mid w \in A \; oder \; w \in B\}$
o	Konkatenation: $A \text{ o B} = \{\mathrm{w} = \mathrm{w}_1 \mathrm{o} \mathrm{w}_2 = w_1 w_2 \mid w_1 \in A \; und \; w_2 \in B\}$
*	Kleene-Stern: $A^* = A^0 \cup A^1 \cup \ldots \cup A^i \cup \ldots$

$\{a\} \text{ o } \{\mathrm{b}\} = \{\mathrm{a}\}\{\mathrm{b}\} = \{\mathrm{ab}\} \quad aber \quad \{a\} \cup \{b\} = \{a, b\}$
$\{a, b\}^* = \{\varepsilon, \mathrm{a}, \mathrm{b}, \mathrm{ab}, \mathrm{ba}, \mathrm{aa}, \mathrm{bb}, \mathrm{aaa}, \mathrm{aab}, \mathrm{aba}, \mathrm{baa}, \mathrm{abb}, \ldots\}$

Definition 4.8:

Σ=Eingabealphabet

Reguläre Ausdrücke über einem Alphabet Σ sind wie folgt induktiv definiert:

a) ∅, ε und **a**, für jedes $a \in \Sigma$, sind reguläre Ausdrücke über Σ und bezeichnen die Mengen ∅, {ε} bzw. {a}.

b) Seien **u** und **v** reguläre Ausdrücke über Σ und $L(\mathbf{u})$ bzw. $L(\mathbf{v})$ die zugehörigen Sprachen. Dann sind auch

1) $(\mathbf{u}+\mathbf{v}) = L(\mathbf{u}) \cup L(\mathbf{v})$,

2) $(\mathbf{u} \cdot \mathbf{v}) = L(\mathbf{u}) \circ L(\mathbf{v})$ und

3) $(\mathbf{u}^*) = L(\mathbf{u})^*$ sowie $(\mathbf{v}^*) = L(\mathbf{v})^*$

reguläre Ausdrücke über Σ.

c) Nur die mit diesen Regeln erzeugten Ausdrücke sind reguläre Ausdrücke über Σ.

Zur Abkürzung schreibt man auch gern $\mathbf{x}^+ = \mathbf{x}\mathbf{x}^*$ und $\mathbf{x}^n = \underbrace{\mathbf{x} \ldots \mathbf{x}}_{n\text{-mal}}$.

Beispiel 4.10:

Die oben betrachtete Sprache $(((\{a\}((\{a\} \cup \{b\})^*))(\{b\}\{b\}))((\{a\} \cup \{b\})^*))$ kann mit folgendem regulären Ausdruck

$(((a * ((a+b)^*)) * (b * b)) * ((a+b)^*))$
Da **Vereinigung** und **Konkatenation** (∪ und ∗) und assoziative Operationen sind, dürfen die betreffenden Klammern einfach weggelassen werden.
$a(a+b)^* bb(a+b)^*$

u, **v** und **w** seien reguläre Ausdrücke.

a) $\emptyset\mathbf{u} = \mathbf{u}\emptyset = \emptyset$ – mit Erinnerung an die Multiplikation mit 0. Sehr wichtig!

b) $\varepsilon\mathbf{u} = \mathbf{u}\varepsilon = \mathbf{u}$ – mit Erinnerung an die Multiplikation mit 1.

c) $\emptyset^* = \varepsilon$

d) $\varepsilon^* = \varepsilon$

e) $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$ – Erinnerung an die Kommutativität der Addition.

f) $\mathbf{u} + \emptyset = \mathbf{u}$ – Erinnerung an die Addition von 0.

g) $\mathbf{u} + \mathbf{u} = \mathbf{u}$ – verstößt gegen die Analogie.

h) $(\mathbf{u}^*)^* = \mathbf{u}^*$

i) $\mathbf{u}(\mathbf{v} + \mathbf{w}) = \mathbf{uv} + \mathbf{uw}$ – mit Erinnerung an die Distributivität.

j) $(\mathbf{uv})^*\mathbf{u} = \mathbf{u}(\mathbf{vu})^*$

k) $(\mathbf{u} + \mathrm{v})^* = (\mathbf{u}^* + \mathbf{v}^*)^*$

Reguläre Sprache $L(G) = \{w \mid w \in \{a,b\}^* \textit{ beginnt oder endet mit } a\}$
Regulärer Ausdruck = $a(a+b)^* + (a+b)^*a$

oder

Reguläre Ausdrücke und endliche Automaten

Eine Sprache L wird von einem $DEA\ M$ genau dann akzeptiert, wenn L durch einen regulären Ausdrucke R über dem zu M gehörenden Eingabealphabet beschrieben werden kann.

$$L(M) = L(R) = L$$

Zu jedem $NEA_\varepsilon M$ gibt es einen äquivalenten $NEA_\varepsilon M'$ mit genau einem Endzustand

$$\mathscr{L}_{NEA_\varepsilon} \equiv \mathscr{L}_{DEA} \equiv \mathscr{L}_{rG}$$