Mengen:

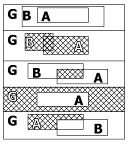
Teilmenge $A \subset B$

Vereinigung $A \cup B$

Durchschnitt $A \cap B$

Komplement $A', \bar{A}$

Komp. von B in A $A \setminus B$



$A \cup \emptyset = A \quad A \cup \overline{A} = G \quad A \cup A = A$

$A \cap G = A \quad A \cap \overline{A} = \emptyset \quad A \cap A = A$

$A \cup B = B \cup A$

$(A \cup B) \cup C = B \cup (A \cup C)$

$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

$A \setminus A = \emptyset \quad A \setminus \emptyset = A$

$A \times \emptyset = \emptyset \times A = \emptyset$

$P(\emptyset) = \{\emptyset\}$

$P(\emptyset) \times P(\emptyset) = \{(\emptyset, \emptyset)\}$

Mächtigkeit einer Menge: $|M| = \text{Anzahl Elemente in } M$

Potenzmenge: Menge aller Teilmengen. Beispiel:

$M = \{a, \psi, \{☺\}\}; \; |M| = 3; \; P(M) = \{\emptyset, M, \{a\}, \{\psi\}, \{\{☺\}\}, \{a, \psi\}, \{\psi, \{☺\}\}, \{\{☺\}, a\}\}$

Mächtigkeit der Potenzmenge: $|P(M)| = \mathbf{2}^{|M|} = 2^3 = 8$

Kartesisches Produkt:

Allgemein: $A \times B = \{(a, b) | a \in A \wedge b \in B\}$

(a, b) heisst "geordnetes Paar"

Bsp: $\{a, b\} \times \{X, Y\} = \{(a, X), (a, Y), (b, X), (b, Y)\}$

Mächtigkeit des Kartesischen Produkts: $|A \times B| = |A| \cdot |B|$

Kartesisches Produkt mit leeren Mengen: $\emptyset \times \emptyset = \emptyset \; ; \; |\emptyset| = 0$

Potenzmenge der leeren Menge : $P(\emptyset) = \{\emptyset\} = \{\{\}\}; \; |P(\emptyset)| = 1$

Kart. Produkt auf $P(\emptyset)$: $P(\emptyset) \times P(\emptyset) = \{(\emptyset, \emptyset)\}$

$P(\emptyset \times \emptyset) = P(\emptyset) = \{\emptyset\}; \quad |P(P(M))| = 2^{2^{|M|}}; \quad P(P(\emptyset)) = \{\emptyset, \{\emptyset\}\}$

Homogene, binäre Relationen: $R \subset M \times M \Rightarrow R \in P(M \times M)$ **//** $xRy :\Leftrightarrow (x, y) \in R$

Anzahl mögliche binäre Relationen R auf eine Menge M : $2^{|M|^2}$

Bei zwei ungleichen Mengen A und B gilt also: $2^{(|A| \cdot |B|)}$

$2^{2^2} = 16; \; 2^{3^2} = 2^9 = 512; \; 2^{4^2} = 2^{16} = 65'536; \; 2^{5^2} = 2^{25} = 33'554'432;$

Anzahl mögliche binäre Relationen auf $P(M)$:

$2^{|P(M)|^2} = 2^{(2^{|M|})^2}$ Bsp: $M = \{1, 2, 3, 4\}; \; 2^{(2^4)^2} = 2^{16^2} = 2^{256} \approx 1.1579 \cdot 10^{77}$

Partitionen: $\Pi_\emptyset = \{\emptyset\}; |\Pi| = Bellzahl$

$\Pi_{\{a,b,c\}} = \{\{\{a, b, c\}\}, \{\{a, b\}, \{c\}\}, \{\{a\}, \{b, c\}\}, \{\{a, c\}, \{b\}\}, \{\{a\}, \{b\}, \{c\}\}\}$

$\Pi_M = \{A_i | i \in \underbrace{I}_{\text{Indexmengen}}\}; \; A_i \neq \emptyset; \; A_i \subset M; \; A_i \cap A_j = \emptyset, \text{falls } i \pm j; \; \underbrace{\cup}_{i \in I} A_i = M$

Bell:
$B_0 = 1$
$B_1 = 1$
$B_2 = 2$
$B_3 = 5$
$B_4 = 15$
$B_5 = 52$
$B_6 = 203$
$B_7 = 877$
$B_8 = 4140$

Funktionen: $f : A \to B$

total: $\forall x \in A \;\; \exists y \in B : (x, y) \in R$ Alle x aus A sind definiert

funktional: $\forall x \in A \; \forall y, z \in B : (x, y) \in R, (x, z) \in R \Rightarrow y = z$

Kein Element aus A hat mehr als einen Partner in B.

surjektiv $:\Leftrightarrow \forall y \in B : \exists x \in A : f(x) = y \; / \; (x, y) \in f$

Jedes Element aus B hat mindestens einen Partner in A.

➡ Alle Zielwerte werden als Funktionswert angenommen.

injektiv $:\Leftrightarrow \forall x, y \in A : f(x) = f(y) \Rightarrow x = y$

Kein Element aus B hat mehr als einen Partner in A.

➡ Alle Zielwerte werden höchstens einmal als Fktwert angenommen.

bijektiv $:\Leftrightarrow f$ ist surjektiv und injektiv

Jedes Element aus B hat genau einen Partner in A ➡ eindeutig umkehrbar.

"trivial": Bsp: $f(x) = 5 // f(x) = x$; Kein mathem. Ausdruck; Hängt von Aufgabe ab

Verkettung: $\mathbf{S} \circ \mathbf{R} := \{(x, z) | \exists y \in M, (x, y) \in \mathbf{R} \wedge (y, z) \in \mathbf{S}\}; (S \circ R)(x) = S(R(x))$

Relationen: $R \subset M^2$:

reflexiv $:\Leftrightarrow \forall x \in M : (x, x) \in R$ Alle mit Loop

irreflexiv $:\Leftrightarrow \forall x \in M : (x, x) \notin R$ Keine Loops

symmetrisch $:\Leftrightarrow \forall x, y \in M : (x, y) \in R \Rightarrow (y, x) \in R$ Doppelpfeile

antisymmetrisch $:\Leftrightarrow \forall x, y \in M : (x, y) \in R \wedge (y, x) \in R \Rightarrow x = y$ Keine Doppelpf.

transitiv $:\Leftrightarrow \forall x, y, z \in M : (x, y) \in R \wedge (y, z) \in R \Rightarrow (x, z) \in R$ Abkürzungen

Äquivalenzrelation: reflexiv, symmetrisch, transitiv. Siehe Partition.

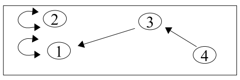
Halbordnung: reflexif, antisymmetrisch, transitiv. Siehe Hasse-Diagramm.

Matrizen: $M := \{1, 2, 3\}; R := \{(1, 1), \mathbf{(2,3)}, (3, 2)\}$

$$A_R := \begin{array}{cc} & (, y) \\ (x,) & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & \mathbf{1} \\ 0 & 1 & 0 \end{pmatrix} \end{array} \quad a_{xy} < \begin{matrix} x \text{ in Zeile,} \\ y \text{ in Spalte} \end{matrix}$$

Anz. Rel: $2^{|M|^2}$ =1,2,16,512...

Graph: $M := \{1, ..., 5\}; \; R = \{(1, 1), (2, 2), (4, 3), (3, 1)\}$



Ungerichteter Graph:

Paar von Mengen $G = (V, K) = (\mathbf{V}\text{ertexmenge}, \mathbf{K}\text{antenmenge})$

V sei eine endliche Menge; $V \neq \emptyset$

K ist eine binäre Rel. auf V; $K \subset V \times V$. K ist irreflexiv und symmetrisch.

Ecken/Vertices v_0, v_1 benachtbart/adjeszent $\Leftrightarrow (v_0, v_1) \in K$

Kanten $(a, b), (c, d)$ benachbart $\Leftrightarrow \{a, b\} \cap \{c, d\} \neq \emptyset$

$v_0, ..., v_r (r > 0)$ heisst **Pfad** $: \Leftrightarrow (v_i, v_{i+1}) \in K$ für $i = 0, ..., r-1$

Pfad heisst **einfach** $\Leftrightarrow v_i \neq v_j$ für $i \neq j$

Pfad heisst **offen** $\Leftrightarrow v_0 \neq v_r$ (kein Ring)

Pfad heisst **geschlossen** $\Leftrightarrow v_0 = v_r$ (Ring)

Länge Pfad = Anzahl Kanten = Anz. Ecken - 1

Länge Zyklus = Anzahl Kanten = Anz. Ecken

Geschl. & einfacher Pfad Z heisst **Zyklus** $\Leftrightarrow$ länge $Z \geq 3$

Graph ohne Zyklen heisst **azyklisch**

Bipartiter Graph: $G = (V_1 \cup V_2, K); \{V_1, V_2\}$ ist Bipartition

Matching: Menge von Kanten, welche nicht benachbart sind.

Maxi<u>males</u> Matching: Keine weiteren Kanten in diesem M. möglich.

Maxi<u>mum</u>-Matching: Es gibt kein grösseres M. Es ist auch maximal.

Hasse-Diagramm: Graphische darstellung endlicher halbgeordneter Mengen.

maximales/minimales Element: Ganz oben/unten

grösstes/kleinstes E.: Alleine oben/unten (auch max./min.)

Halbordnung: reflexif, antisymmetrisch, transitiv.

Ordnungshomomorphismus (ordnungstreue Abbildung):

Gegeben: $(T(30), |)$ und $(P(\{a, b, c\}), \subset)$ seien zwei Halbordnungen.

Eine Abbildung $h : T(30) \to P(\{a, b, c\})$ heisst O. $\Leftrightarrow \forall x, y \in T(30) : (x | y \Rightarrow h(x) \subset h(y))$

Ordnungsisomorph: Bijektiver Homomorphismus.

Aussagenlogik (AL):

Literal: $p_1, (\neg(p_1))$ Variable oder negative Variable

L$_{AL}$: {Formeln über AL }

A$_{AL}$: $AV \cup LJ \cup HZ = \{p_1, p_2, ...\} \cup \{\wedge, \vee, \neg, \to, ...\} \cup \{(,)\}$

Klausel: $\varphi = L_1 \vee L_2 \vee ... \vee L_r$ mit L_i Literale und $r \in \mathbb{N}$

Konjunktive Normalform: $\psi \in KNF \Leftrightarrow \psi = \varphi_1 \wedge \varphi_2 \wedge ... \wedge \varphi_s; \; s \geq 1 \wedge \varphi_i$ Klauseln

1) Wenn eine Klausel = 0, dann KNF = 0

2) Wenn alle K. = Tautologie, dann KNF = Tautologie

3) Wenn eine K. = Kontradiktion, dann KNF = Kontr.

Var1	Var2	und	oder	folgt	äquiv.	xor
p_1	p_2	$\wedge$	$\vee$	$\to$	$\leftrightarrow$	$\oplus$
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	0	0	1
1	1	1	1	1	1	0

Äquivalente Formeln: $\vDash (\varphi \leftrightarrow \psi)$

$\varphi \vee (\psi \wedge \varrho) \approx (\varphi \vee \psi) \wedge (\varphi \vee \varrho)$	$\varphi \wedge (\psi \vee \varrho) \approx (\varphi \wedge \psi) \vee (\varphi \wedge \varrho)$
$\neg(\varphi \vee \psi) \approx (\neg\varphi) \wedge (\neg\psi)$	$\neg(\varphi \wedge \psi) \approx (\neg\varphi) \vee (\neg\psi)$
$\neg(\neg\varphi) \approx \varphi$	$\varphi \vee (\neg\varphi) \approx 1 \; ; \; \varphi \wedge (\neg\varphi) \approx 0$
$\varphi \wedge (\varphi \vee \psi) \approx \varphi$	$\varphi \vee (\varphi \wedge \psi) \approx \varphi$
$\varphi \wedge ((\neg\varphi) \vee \psi) \approx \varphi \wedge \psi$	$\varphi \vee ((\neg\varphi) \wedge \psi) \approx \varphi \vee \psi$
$(\varphi \wedge \psi) \vee (\varphi \wedge (\neg\psi)) \approx \varphi$	$(\varphi \vee \psi) \wedge (\varphi \vee (\neg\psi)) \approx \varphi$
$\varphi \to \psi \approx ((\neg\varphi) \vee \psi)$	$(\neg\varphi) \to (\neg\psi) \approx \varphi \vee (\neg\psi) \approx \psi \to \varphi$
$(\neg\varphi) \to \psi \approx (\varphi \vee \psi)$	$\varphi \to (\neg\psi) \approx (\neg(\varphi \wedge \psi)) \approx (\neg\varphi) \vee (\neg\psi)$
$\neg(\varphi \to (\neg\psi)) \approx (\varphi \wedge \psi)$	$(\varphi \leftrightarrow \psi) \approx ((\varphi \wedge \psi) \vee ((\neg\varphi) \wedge (\neg\psi)))$
$(\neg(\varphi \to \psi)) \approx \varphi \wedge (\neg\psi)$	$(\varphi \oplus \psi) \approx (((\neg\varphi) \wedge \psi) \vee (\varphi \wedge (\neg\psi)))$

Ein Modell ist falsifizierbar/erfüllbar, Tautologie/Kontradiktion.

Modell: $\vDash_w \varphi$ // "w ist ein Modell für φ"

Logische (/Semantische) **Folgerung**: $\psi_1, \psi_2, ... \psi_n \vDash \varphi$

Definition: φ folgt semantisch aus $\psi_1, \psi_2, ...$

$\psi_1, \psi_2 \vDash \varphi$ ist wahr, wenn $\forall \vDash_\omega \psi_n \Rightarrow \vDash_\omega \varphi$.

Belegung:

$\omega : AV \to \{0, 1\}$

Bsp: $\omega(p_1) = 1; \omega(p_2) = 0;$

Interpretation:

$I_\omega : L_{AL} \to \{0, 1\}$

Bsp: $I_\omega(p_1 \wedge p_2) = 0$

Sprachen:

Alphabet: $\Sigma =$ Menge von endlichen Zeichen. Bsp: $\{a, b, c\}$

Wörter: $\omega \in \Sigma^*$: Menge aller endlichen Wörter über Σ .

Wortlänge: $|\omega|$ = Anzahl Zeichen; **Leeres Wort:** ε ; $|\varepsilon| = 0$

Sprachen über Σ : $P(\Sigma^*)$: Menge aller Sprachen über Σ

Sprache: $L \subset \Sigma^*$; Teilmenge von Wörtern aus Σ^*

Beschreibung von Spr.: Aufzählen, Generieren, Akzeptieren, Konstruieren

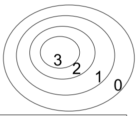
Grammatik: 4-Tupel $G := (N, T, R, S)$

N : Nichtterminalalphabet (Variabeln); T : Terminalalphabet;

S : Startsymbol; R : Regeln $R \subset (N \cup T)^+ \times (N \cup T)^*$

$\Rightarrow$ / $\Rightarrow^*$: Anwendung einer/mehrerer Regeln

$\boldsymbol{L}(\mathbf{G}) := \{w \in T^* | S \Rightarrow^* w\}$ Sprache die aus G generiert wird.



Chomsky-Hierarchie: G nach Regeln klassifizieret

Typ	Bezeichnung	Bedingung für Regel: $u \to v$	Art: $\alpha, \beta, \gamma \in (N \cup T)^*$ $A, B \in N \wedge a \in T$
0	allgemein	$-$	*beliebig*
1	kontextsensitiv	$u = \omega_1 A \omega_2$ | $\omega_1, \omega_2, \omega \in (N \cup T)^*$ $v = \omega_1 \omega \omega_2$ | $\omega \neq \varepsilon, A \in N$ $\|u\| \leq \|v\| \Rightarrow nichtverkürzend$	$\alpha A \beta \to \alpha \gamma \beta$
2	kontextfrei	$u \in N, v \in (N \cup T)^*$ wie Typ 1 mit $\omega_1, \omega_2 = \varepsilon$	$A \to \gamma$
3 "beste"	regulär	$u \in N, v = a \in T \vee v = aA, A \in N$ $\|u\| = 1; \; 2 \geq \|v\| \geq 1$	$A \to a \| aB$ (rechtsregulär) $A \to a \| Ba$ (linksregulär)

Die Regel $S \to \varepsilon$ steht "ausser Konkurenz" ➡ keine Relevanz bei Klassifizierung!

Klasse einer Grammatik G : "schlechteste" Regel aus Grammatik G .

Klasse einer Sprache $L(G)$: Klasse einer bestmöglichen Gram. die L erzeugt.

Notation: $L \in \mathscr{L}_i \Rightarrow$ Sprache L ist vom Typ i . D.h. es gibt eine Typ-i-Grammatik welche L erzeugt. $\mathscr{L}_i$ Ist die Menge aller Sprachen vom Typ i

Operationen auf Sprachen von $P(\Sigma^*)$:

Sei Σ ein Alphabet und $\mathscr{L} = P(\Sigma^*)$ ($\mathscr{L}$ =Menge der Sprachen über Σ)

Seien L_1 und $L_2 \in \mathscr{L}$

Bekannte Operationen: $\overline{L_1}$ (Komplement); $L_1 \cup L_2$; $L_1 \cap L_2$; $L_1 \setminus L_2$; ...

Konkatenation: $L_1 \circ L_2 = \{\omega_1 \omega_2 | \omega_1 \in L_1, \omega_2 \in L_2\}$ $\quad |L_1 \circ L_2| = |L_1| \cdot |L_2|$

Stern- oder Kleene-Operator: $L^* = L^0 \cup L^1 \cup L^2 \cup ... \cup L^n$

$L^0 = \{\varepsilon\}; \; L^1 = L; \; L^2 = L^1 \circ L; \; L^2 = L^2 \circ L = (L \circ L) \circ L; \; L^n = L^{n-1} \circ L$

Entprechend mit Alphabet: $\Sigma^* = \{\varepsilon\} \cup \Sigma \cup \Sigma \circ \Sigma \cup \Sigma \circ \Sigma \circ \Sigma \cup ... \cup \Sigma^n; \quad \Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$

Als Partition: $\Sigma^* = \bigcup_{n \in \mathbb{N}_0} \{\omega \mid |\omega| = n\}$ (Alle Wörter mit Länge von 0 bis n)

Reguläre Ausdrücke:

Ausdruck	PCRE	Wörter (generierte Sprache)	Operation
a	/a/	{a}	-
a*	/a*/	{ε,a,aa,aaa,aaaa,...}	Kleene-Stern
[ab]	/ab/	{a}○{b} = {ab}	Konkatenation
[aUb]	/(a|b)/	{a}U{b} = {a,b}	Vereinigung
[a*U[bc]*]	/(a*|(bc)*)/	{ε,a,aa,aaa,...} U {ε,bc,bcbc,bcbcbc,...}	(*Kombination*)
[εUb]	/b?/	{ε}U{b} = {ε,b}	Vereinigung

Syntax: definiert, wie reguläre Ausdrücke auszusehen haben: regex = [a*Ub*]

Semantik: definiert, welche formale Bedeutung die Syntax hat: E(regex) = {a}*U{b}*

Grammatik:

$G_{reg} = (N, T, R, S); N = \{S\};$

$\mathcal{A} = \{a_1, a_2, ..., a_n\}$

$T = \mathcal{A} \cup \{\emptyset, \varepsilon, \cup, *, [,]\}$

$R = \{ \;\; \forall b \in \mathcal{A} \cup \{\emptyset, \varepsilon\} : S \to b,$

$S \to [S \cup S], \; S \to [SS], \; S \to S^* \}$

E (Kleene-Mykill): $E : (Reg_{\mathcal{A}}) \to \mathscr{Reg}_{\mathcal{A}}$

$E(\text{Ausdruck}$ (=Zeichenkette)$) := \text{Sprache}$ (=Menge von Wörtern)

$E(\emptyset) := \emptyset$

$E(\varepsilon) := \{\varepsilon\}$

$\forall a \in A : a \to E(a) := \{a\}$

$\forall \alpha, \beta \in Reg_{\mathcal{A}}$

$E([\alpha \cup \beta]) := (E(\alpha) \cup E(\beta))$

$E([\alpha \beta]) := (E(\alpha) \circ E(\beta))$

$E(\alpha^*) := E(\alpha)^*$

Korollar: Jede reguläre Sprache über $\mathcal{A}$ lässt sich aus den endlichen Sprachen über $\mathcal{A}$ mithilfe der Operationen $\cup$, $\circ$, und $*$ konstruieren.

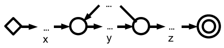
Pumping-Lemma:

Sei $L \in \mathcal{R}eg_\Sigma \Rightarrow \exists A \in DFA: L(A)=L$ d.h. $A=(Q,\Sigma,\delta,q_0,F); Q=\{q_0,\ldots,q_k\}; n=|Q|; |L|=\infty$

Sei $\omega \in L$ mit $|\omega| \geq n$. $\omega = xyz$; $|xy| \leq n$; $|y| \geq 1$; $\forall k \in \mathbb{N}: x y^k z \in L$

Da ein Zustand mehr als einmal durchquert wird, kann y beliebig oft vorkommen.

Bsp: $L \in P(\{0,1\}^*); L=\{0^k,1^k\}; k \in \mathbb{N}; n \geq 1$

$\omega = \underbrace{\overbrace{0\ldots0}^{k}}_{xy}\underbrace{\overbrace{1\ldots1}^{k}}_{z} \in L$ $//|\omega_k|=2k$ $\Rightarrow y=0\ldots0$ $\Rightarrow \tilde{\omega}=xyyz \in L$ 

Automaten:

Deterministischer endlicher Automat (DFA): 5-Tupel $A=(Q,\Sigma,\delta,q_o,F)$

Q : Zustände; Σ : Alphabet; δ : $Q \times \Sigma \to Q$ (Zustandsübergangsfunktion)

q_0 : Anfangszustand; $F \subset Q$: Akzeptierte Endzustände

Von jedem Zustunde gehen so viele Pfeile weg wie es Buchstaben in Σ hat!

Antwortfunktion: $\boldsymbol{r_A: \Sigma^* \to Q} \Leftrightarrow r_A(\varepsilon)=q_0; \ \forall \omega \in \Sigma^*, a \in \Sigma: r_A(\omega a)=\delta(r_A(\omega),a)$

$\boldsymbol{L(A)}:=\{\omega \in \Sigma^* | r_A(\omega) \in F\}$ die vom Automaten A akzeptierte (reguläre) Sprache.

Startzust.: ◇ **akzeptierter Z:** ◎ **übrige Z:** ○ **akz. Start-Z.:** ◈ $\Rightarrow \varepsilon \in L(A)$

Nichtdeterministische Automaten (NFA):

Der Automat hat auf dem Weg zum Ergebnis Wahlfreiheiten.

$\forall A \in \text{NFA} \ \exists B \in \text{DFA} : L(A)=L(B)$ mit $\delta_B: Q \times \Sigma \to P(Q)$

Determinisierung nach Rabin/Scott (Umwandlung NFA → DFA):

$A=(Q_A,\Sigma,\delta_A,q_{0A},F_A); B=(Q_B,\Sigma,\delta_B,q_{0B},F_B); L(A)=L(B)$

$Q_B=P(Q_A); \quad q_{0B}=\{q_{0A}\};$

R: Zustand von B, also $R \in P(Q_A)$: $\delta_B(R,a)=\{q \in Q_A | \exists r \in R: \delta_A(r,a)=q\};$

Anders gesagt: $\delta_B(R,a)=\bigcup_{r \in R} \delta_A(r,a)$

$F_B=\{R \in P(Q_A) | R \cap F \neq \emptyset\}$

Beispiel:

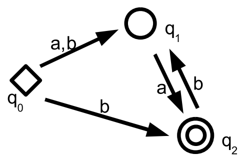
$A=(Q_A,\Sigma,\delta_A,q_0,F_A); B=(Q_B,\Sigma,\delta_B,\{q_0\},F_B);$

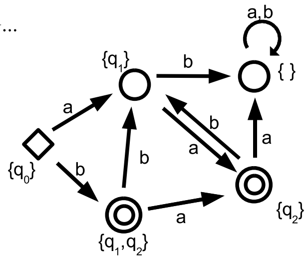
$Q_A=\{q_0,q_1,q_2\}; \ \Sigma=\{a,b\}; \ F_A=\{q_2\};$

$Q_B=P(Q_A)=\{\emptyset,\{q_0\},\{q_0,q_1\},\ldots\};$

$\delta_A(q_0,a)=q_1; \delta_A(q_0,b)=q_1|q_2\ldots$

$\delta_B(\{q_0\},a)=\{q_1\}; \delta_B(\{q_0\},b)=\{q_1,q_2\}\ldots$

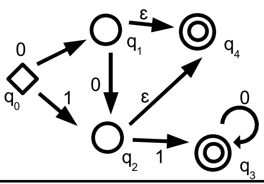


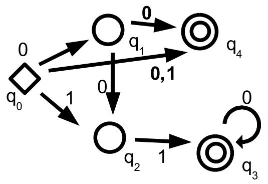


NFA/ε:

Ohne weiteren Buchstaben zu lesen, kann der Zustand durch ε geändert werden.

Dieser wird zuerst in einen NFA (ohne ε) gewandelt und anschliessend in einen DFA.



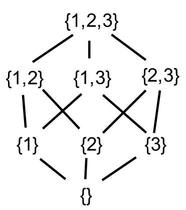


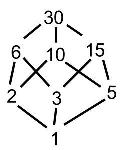
Sonstiges/Beispiele:

Primzahlen: 2 | 3 | 5 | 7 | 11 | 13 | 17 | 19 | 23 | 29 | 31 | 37 | 41 | 43 |
| 47 | 53 | 59 | 61 | 67 | 71 | 73 | 79 | 83 | 89 | 97 | (1 gehört per Def. nicht dazu!)

T(32) = {1,2,4,8,16,32}; T(36) = {1,2,3,4,6,9,12,18,36}; T(34) = {1,2,17,34};
T(30) = {1,2,3,5,6,10,15,30}
P({1,2,3})

Für $(P(\{1,2,3\}),\subseteq)$ und $(T(30),|)$ sind ordnungsisomorph:



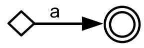


{({1,2,3},30), ({1,2},6), ({1,3},10), ({2,3},15), ({1},2), ({2},3), ({3},5), ({},1)};

Spickzettel "Diskrete Mathematik 1" für MSP.
Erstellt von Claude Martin. FHNW Windisch. Studiengang Informatik.
Herbstsemester 2010/2011. Dozent: D. Mall.
Alles ohne Gewähr! Kein Anspruch auf Korrektheit oder Vollständigkeit!

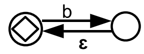
Regexp -> Automat:

Automat für Regexp "a":

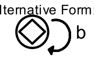
$A(\text{a})$: 

Kleene Operation auf einen Automaten: $A(\text{b}^*)=A(\text{b})^*$

◎ von A werden durch ε mit dem ◇ (als ◈) von A verbunden.



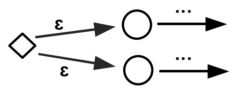
Alternative Form:



$A([\text{a} \cup \text{b}])=A(\text{a}) \cup A(\text{b})$

Verkettung zweier Automaten: $\text{A} \cup \text{B}$

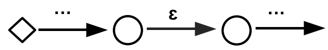
Neues ◇ wird durch ε jeweils mit ◇ von A und B (als ○) verbunden.



$A([\text{ab}])=A(\text{a}) \circ A(\text{b})$

Verkettung zweier Automaten: $\text{A} \circ \text{B}$

◎ von A werden durch ε mit dem ◇ von B (als ○) verbunden.



Spezielle Reguläre Ausdrücke:

Leere Sprache: { }

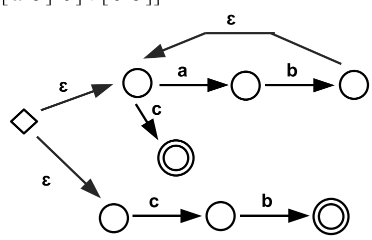
$\emptyset \to A(\emptyset)$: 

Sprache mit leerem Wort: **{ε}**

$\varepsilon \to A(\varepsilon)$: 

Ein Beispiel:

$[[[\text{a b}]^* \text{c}] \cup [\text{c b}]]$



Sprache:

$E(\ [[[\text{a b}]^* \text{c}] \cup [\text{c b}]] \ ) = \{c, abc, ababc, abababc, \ldots\} \cup \{cb\}$

Notizen: