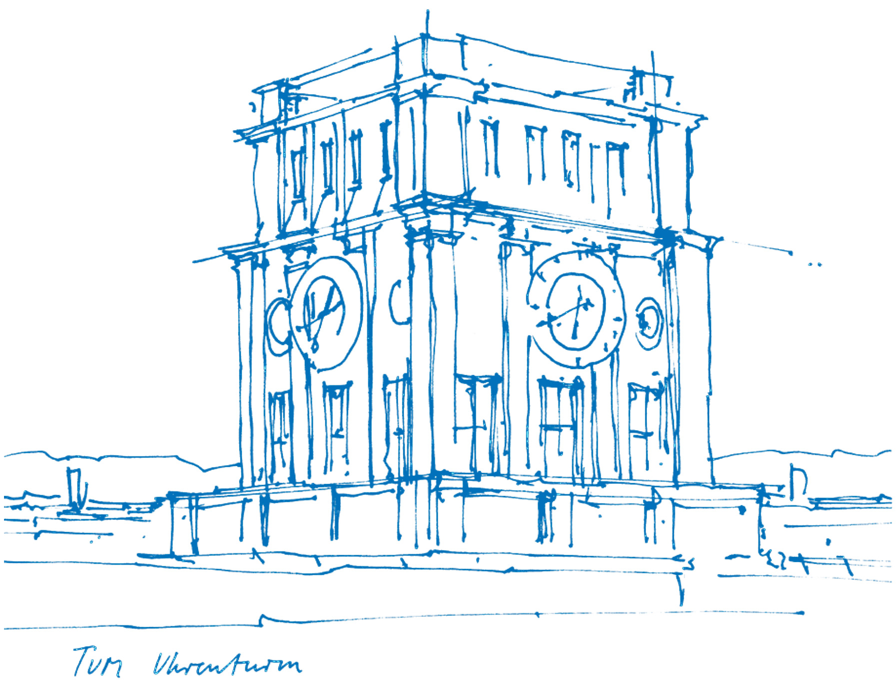
Diskrete Strukturen Tutorium

Jay Zhou
Technische Universität München
Garching b. München, 6. November 2023



Relationen — Äquivalenzrelationen

Reflexiv

$\mathrm{Id}_A \subseteq \equiv_f$

Symmetrisch

$a \equiv_f b \iff f(a) = f(b) \iff b \equiv_f a$

Transitiv

$(m_1, m_2) \in \equiv_f$ und $(m_2, m_3) \in \equiv_f \iff (m_1, m_3) \in \equiv_f$

aka. $m_1 \equiv_f m_3$

Relationen — Äquivalenzrelationen

Repräsentantensystem

- Eine Teilmenge, die für jede Äquivalenzklasse genau ein Element aus dieser Klasse hat

repräsentieren

Beispiel

- mod 3: $\{0, 1, 2\}$ 3 4 5 ...
- $L((ab)\Sigma^*)$: $\{[\epsilon], [a], [ab], [b]\}$

Relationen — Ordnungsrelation

Partielle Ordnung

− Reflexiv, antisymmetrisch und transitiv

Totale Ordnung

− Partielle Ordnung

− Je zwei beliebige a, b stets bzgl. R in Relation stehen (mindestens aRb oder bRa gilt)

Relationen — Ordnungsrelation

Reflexiv

$\mathrm{Id}_A \subseteq \equiv_f$

Antisymmetrisch

$\forall m_1 \neq m_2 \,.\, (m_1, m_2) \in \equiv_f \to (m_2, m_1) \notin \equiv_f$

Transitiv

$(m_1, m_2) \in \equiv_f$ und $(m_2, m_3) \in \equiv_f \iff (m_1, m_3) \in \equiv_f$

aka. $m_1 \equiv_f m_3$

Relationen — Ordnungsrelation

Hasse Diagram

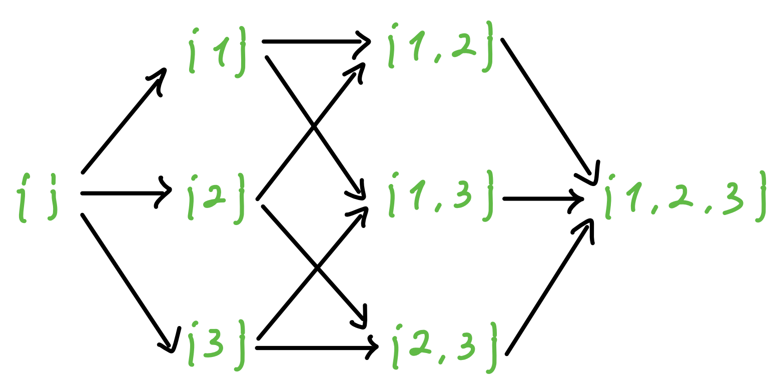
− eine bestimmte graphische Darstellung endlicher halbgeordnete Menge

− Jede partielle Ordnung über einer Menge kann man ein eindeutiges Hasse Diagram zuordnen.

Quiz: Hasse Diagram über Teilmenge auf $P([3])$? Und über Teilbarkeit auf 60?

$\{\}, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}$

$$\{\} \preceq \begin{array}{c}\{1\}\\ \{2\}\\ \{3\}\end{array} \preceq \begin{array}{c}\{1,2\}\\ \{1,3\}\\ \{2,3\}\end{array} \preceq \{1,2,3\} \quad \Rightarrow$$



Relationen — Ordnungsrelation

Hasse Diagram

− eine bestimmte graphische Darstellung endlicher halbgeordnete Menge

− Jede partielle Ordnung über einer Menge kann man ein eindeutiges Hasse Diagram zuordnen.

Quiz: Hasse Diagram über Teilmenge auf $P([3])$? Und über Teilbarkeit auf 60?

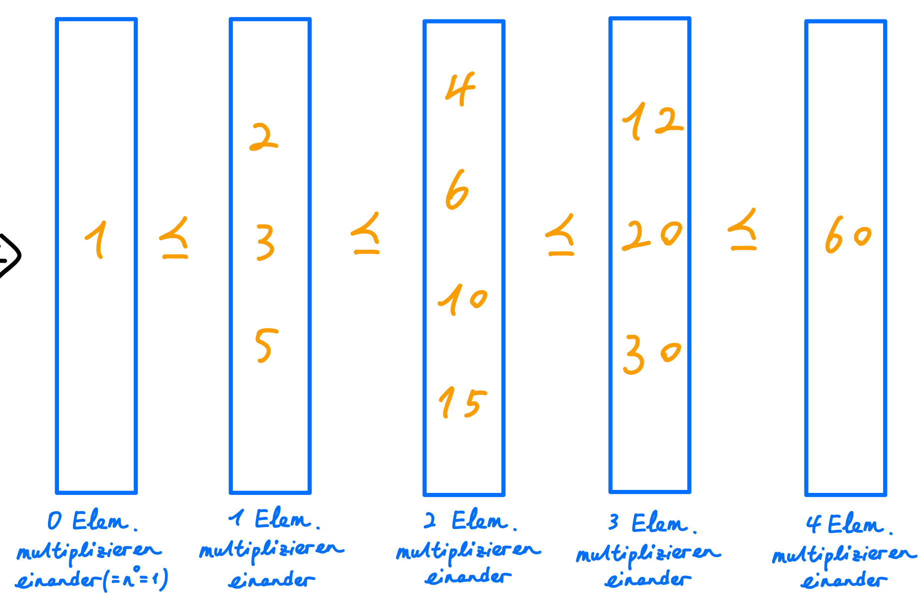
Primfaktorzerlegung

$$\begin{array}{r|l} 2 & 60 \\ 2 & 30 \\ 3 & 15 \\ & 5 \end{array} \quad = 2^2 \cdot 3 \cdot 5$$

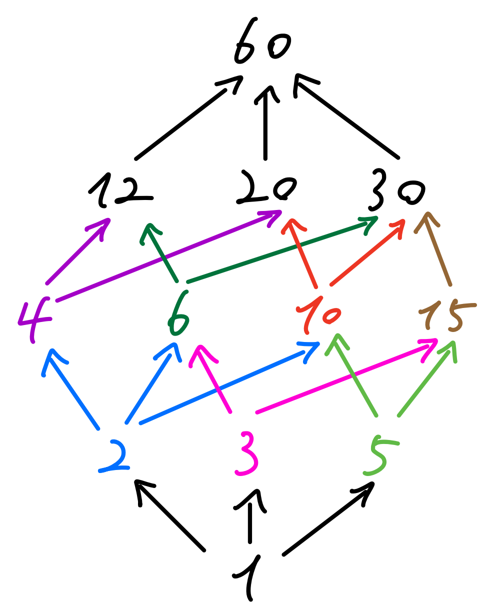
4 Stammelemente

2 , 2 , 3 , 5

$\Rightarrow$



$\Rightarrow$



Voraussetzung von $a \to b$: $a \leq b$

Wegen partiellen Ordnung : reflexiv $(a,a) \in \preceq$

transitiv $(a,b),(b,c) \in \preceq \Rightarrow (a,c) \in \preceq$

antisymmetrisch $(a,b) \in \preceq \Rightarrow (b,a) \notin \preceq$ Falls $a \neq b$

Aufgabe 3.1

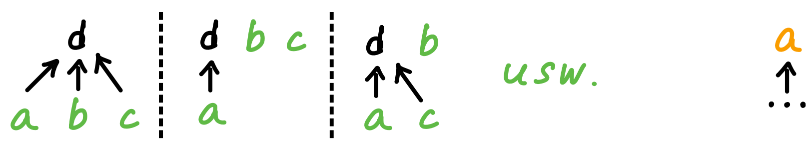
Geben Sie die Hasse-Diagramme aller partiellen Ordnungen $\preceq$ auf $[4]_0 := \{0, 1, \dots, 4\}$ an, welche die folgenden beiden Bedingungen erfüllen:

- Falls $a \preceq b$ gilt, dann auch $a \leq b$.

  (Mit anderen Worten: $\{(a,b) \in [4]_0 \times [4]_0 \mid a \preceq b\} \subseteq \{(a,b) \in [4]_0 \times [4]_0 \mid a \leq b\}$.)
- Bzgl. $\preceq$ gibt es genau 3 minimale Elemente und ein größtes Element.

Wie verändert sich die Anzahl, wenn man isomorphe Hasse-Diagramme nur einmal zählt?

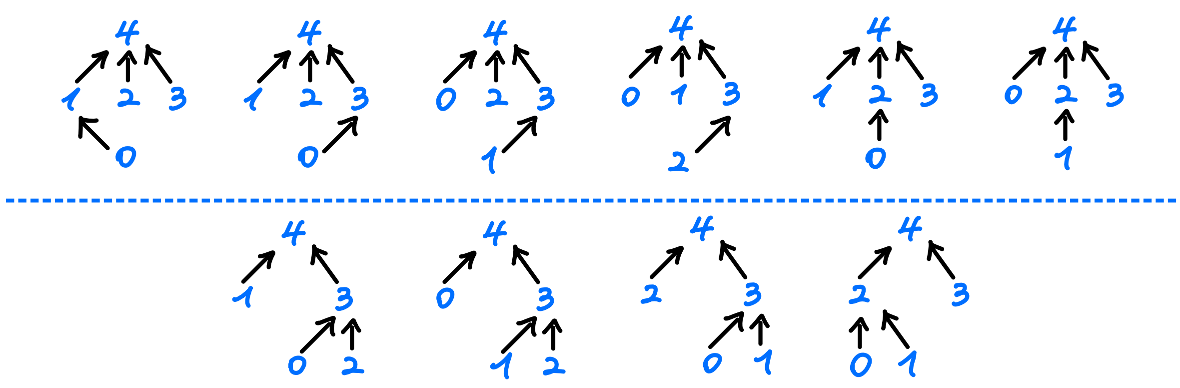
Hierarchien-struktur

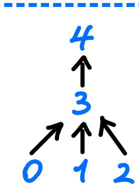


Struktur 1-3-1

Struktur 1-2-2

Struktur 1-1-3



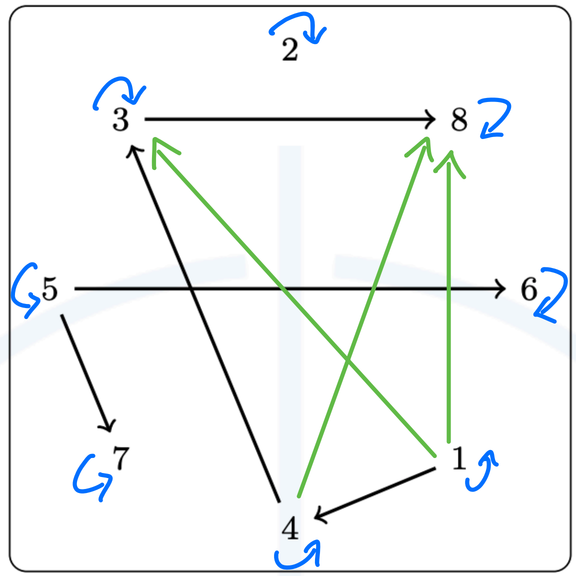


$\Rightarrow$ Bis auf Isomorphie 3 Klassen !

$\Rightarrow$ Insgesamt 11

Aufgabe 3.2

Gegeben ist die folgende Relation R über der Grundmenge $[8]$ in graphischer Darstellung:



hinzufügen,
nur wenn notwendig

(a) Bestimmen Sie die kleinste partielle Ordnung S mit $R \subseteq S$, soweit diese existiert, andernfalls begründen Sie, warum es keine solche Relation gibt.

reflexiv $\quad \forall n \in [8].\ Id_n \in \preceq$

transitiv $\quad (1,4) \circ (4,3) = (1,3)$

$(4,3) \circ (3,8) = (4,8)$

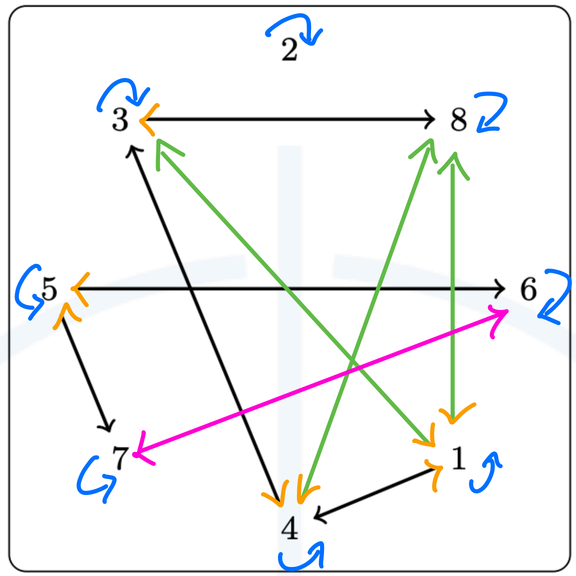
$(1,4) \circ (4,8) = (1,8)$

antisymmetrisch ✓

Was passiert, wenn es im Graph bereits einen symmetrischen Paar gibt? (Bsp. $(1,2), (2,1) \in \preceq$)
-- Eine solche partielle Ordnung ist nicht möglich.

Aufgabe 3.2

Gegeben ist die folgende Relation R über der Grundmenge $[8]$ in graphischer Darstellung:



hinzufügen, nur wenn notwendig

(b) Bestimmen Sie die kleinste Äquivalenzrelation T mit $R \subseteq T$, soweit diese existiert, andernfalls begründen Sie, warum es keine solche Relation gibt.

reflexiv $\quad \forall n \in [8] .\ Id_n \in \preceq$

transitiv $\quad (1,4) \circ (4,3) = (1,3)$

$(4,3) \circ (3,8) = (4,8)$

$(1,4) \circ (4,8) = (1,8)$

symmetrisch $\quad \forall (a,b) \in \preceq .\ (b,a) \in \preceq$

Wichtig: am Ende nochmal auf Transitivität prüfen! Da neue transitive Relationen durch Symmetrie erzeugt werden.

$(6,5) \circ (5,7) = (6,7)$ und $(7,6)$ durch Symmetrie

Aufgabe 3.3

Sei $\Sigma = \{a, b, c\}$ ein Alphabet. Wir betrachten die folgende Sprache $L \subseteq \Sigma^*$:

$$L = \{w \in \Sigma^* \mid w \text{ enthält sowohl ein } a \text{ als auch ein } b\}$$

Bezüglich L definieren wir die Äquivalenzrelation $\sim_L$ auf Σ^* durch

$$u \sim_L v \text{ falls für alle } z \in \Sigma^* \text{ gilt: } uz \in L \text{ genau dann, wenn } vz \in L$$

Bestimmen Sie alle 4 Äquivalenzklassen von $\sim_L$. Geben Sie auch ein Repräsentantensystem an, das möglichst kurzen Wörtern besteht.

Gesucht: 4 Klassen von Wörtern,
jeweils eine Vertretung => kleinste Wort

Beispiel: $u = c$, $v = cc$

Sei $z = \varepsilon$	Sei $z = a$	Sei $z = ab$
$uz = c$	$uz = ca$	$uz = cab$
$vz = cc$	$vz = cca$	$vz = ccab$
$uz \sim_L vz$	$uz \sim_L vz$	$uz \sim_L vz$

2 Verfahren
- Intuition ★ bevorzugt : Musterlösung
- Automat : NICHT VL-STOFF, nur zum Exkurs.
  (Stoff aus IN0014 – Theoretische Informatik) Schwer, aber garantiert richtig :)

① Intuition

Mögliche u und v ? Da die Wörter enthalten sowohl a als auch b ...

– Idee : Betrachtet man die Reihenfolge der qualifizierten Wörtern.
Nach einem a müsste es einen b geben (auch umgekehrt).
Wir schneiden die Wörtern in $\underbrace{\text{2 Teile}}_{w = uz}$, beobachten $\underbrace{\text{den ersten}}_{u}$ und bekommen die folgende

– Klasse 1 : Alle Wörter, die weder a noch b enthalten

Falls $a \in z$ und $b \in z$ $\Rightarrow$ $uz \sim_L vz$, da $uz \in L$ und $vz \in L$

Sonst $\Rightarrow$ $uz \sim_L vz$, da $uz \notin L$ und $vz \notin L$

– Klasse 2 : Alle Wörter, die a enthalten , aber nicht b

Falls $b \in z$ $\Rightarrow$ $uz \sim_L vz$, da $uz \in L$ und $vz \in L$

Sonst $\Rightarrow$ $uz \sim_L vz$, da $uz \notin L$ und $vz \notin L$

– Klasse 3 : Alle Wörter, die b enthalten , aber nicht a

Falls $a \in z$ $\Rightarrow$ $uz \sim_L vz$, da $uz \in L$ und $vz \in L$

Sonst $\Rightarrow$ $uz \sim_L vz$, da $uz \notin L$ und $vz \notin L$

– Klasse 4 : Alle Wörter, die a und b enthalten

Es gilt stets $\Rightarrow$ $uz \sim_L vz$, da $uz \in L$ und $vz \in L$

- Klasse 1: Alle Wörter, die weder a noch b enthalten

  $\mathcal{L} = c^*$

  Wortvertretung (kleinstes Wort): ε

- Klasse 2: Alle Wörter, die a enthalten, aber nicht b

  $\mathcal{L} = \{a, c\}^* a \{a, c\}^*$

  Wortvertretung: a

- Klasse 3: Alle Wörter, die b enthalten, aber nicht a

  $\mathcal{L} = \{b, c\}^* b \{b, c\}^*$

  Wortvertretung: b

- Klasse 4: Alle Wörter, die a und b enthalten

  $\mathcal{L} = \Sigma^* a \Sigma^* b \Sigma^* \cup \Sigma^* b \Sigma^* a \Sigma^*$

  Wortvertretung: ab oder ba

$\Rightarrow$ Repräsentantensystem

$[\varepsilon] = \{c\}^*$

$[a] = \{a, c\}^* a \{a, c\}^*$

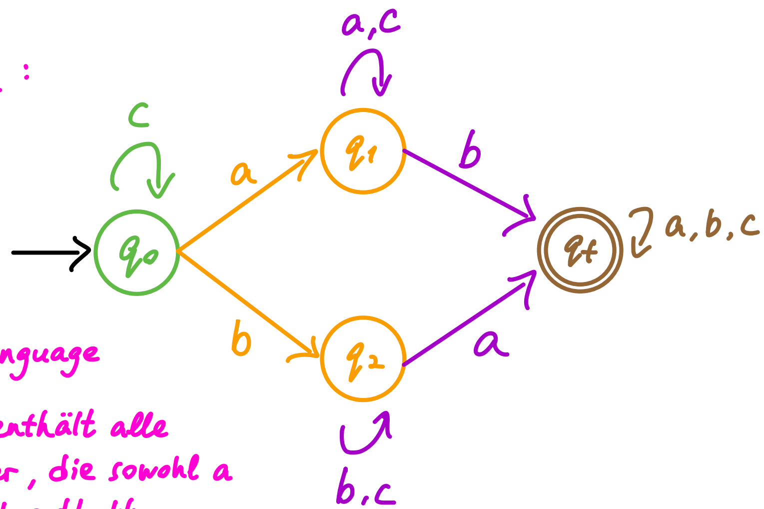
$[b] = \{b, c\}^* b \{b, c\}^*$

$[ab]$ (oder $[ba]$) $= \Sigma^* a \Sigma^* b \Sigma^* \cup \Sigma^* b \Sigma^* a \Sigma^*$

② Automat – DFA (Deterministic Finite Automata)

$\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$
$Q = \{q_0, q_1, \dots, q_f\}$ Zustandmenge
$\Sigma = \{a, b, \dots\}$ Alphabeten
δ : Übergänge q_0 : Anfangszustand
F : Menge von Endzustände

$\mathcal{M}$:



$\mathcal{L}$ = language

$\mathcal{L}(\mathcal{M})$ enthält alle Wörter, die sowohl a auch b enthalten.

Idee : Das Wort wird von links nach rechts gelesen, die Zustände werden entsprechend geschalten.

Die Wörter, die den Endzustand erreichen, werden akzeptiert.

Achtung : Für jeden Zustand sollte es eine Schaltung für jeden Alphabet geben. Überwindung : Garbage-Zustand $\rightarrow \bigcirc \supseteq \Sigma$

In diesem Zustand wird das Wort gerade ins Automat geworfen. Da „c" keinen Einfluss auf die von uns betrachtete Wortmenge hat (w enthält sowohl b als auch a), wird der Zustand bei der Eingabe von c nicht geändert.

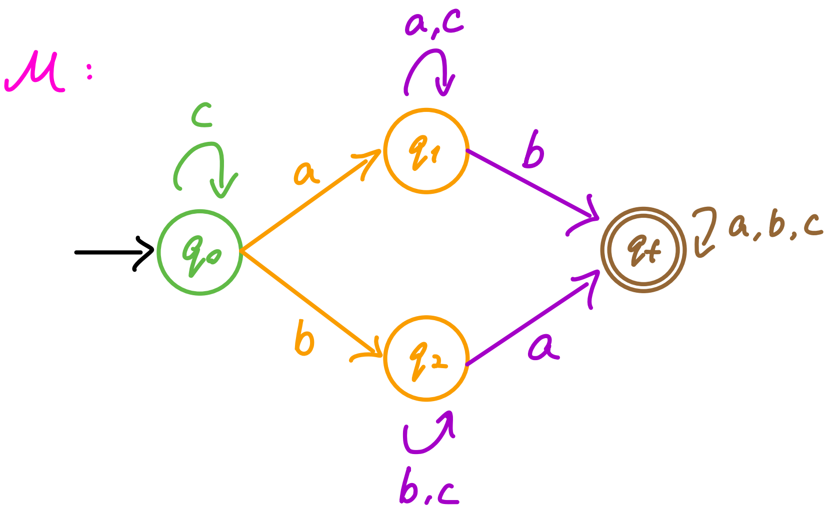
Da die Wörter, die wir betrachten, sowohl b als auch a enthalten, ist ein Zustandwechsel nach der Eingabe von a oder b erforderlich. Beispielsweise wird es nach der Eingabe von a erkannt, dass das Wort a enthält. Es fehlt also hier noch ein b. Deswegen setzen wir in einen Zustand fort, wo das Wort bereits a enthält, aber noch einen b benötigt. Somit bleibt das Wort in diesem Zustand, wenn weitere as und cs eingegeben werden. Dieselbe Prinzip gilt auch für b.

Wir beobachten nun den oberen Pfad. Hier brauchen wir noch ein b. Deswegen wird der Zustand bei der Eingabe eines bs gewechselt. Somit enthält das Wort beide a und b. Für den unteren Pfad gilt es auch, aber mit der Eingabe von a stattdessen.

Der letzte Zustand beschreibt alle Wörter, die sowohl a, als auch b enthalten. Eine beliebige Erweiterung ist daher möglich. Soeben darf das Wort zu einem beliebigen Zeitpunkt halten.

② Automat - DFA (Deterministic Finite Automata)

$\mathcal{M}$:



$\mathcal{L}(\mathcal{M})$ enthält alle Wörter, die sowohl a auch b enthalten.

	Sprache	Kleinstes Wort
q_0	c^*	ε
q_1	$c^* a \{a,c\}^*$	a
q_2	$c^* b \{b,c\}^*$	b
q_f	$c^* a \{a,c\}^* b \Sigma^* \cup c^* b \{b,c\}^* a \Sigma^*$	ab ODER ba

$\Rightarrow$ Repräsentantensystem

$[\varepsilon] = \{c\}^*$

$[a] = \{c\}^* a \{a,c\}^*$

$[b] = \{c\}^* b \{b,c\}^*$

$[ab]$ (oder $[ba]$) $=$

$\{c\}^* a \{a,c\}^* b \Sigma^*$

$\cup \{c\}^* b \{b,c\}^* a \Sigma^*$

* Unterschied zwischen den Sprachen?

① Beweis von $\underbrace{\mathcal{L}(\{a,c\}^* a \{a,c\}^*)}_{\mathcal{L}_1} = \underbrace{\mathcal{L}(\{c\}^* a \{a,c\}^*)}_{\mathcal{L}_2}$

$\supseteq$: Trivial

$\subseteq$: Falls das Wort mit a anfängt, wird das a zum mittleren a beim $\mathcal{L}_2$ zugeordnet. Was im $\mathcal{L}_1$ übrig bleibt, enthalten nur a und c, ist also eine Teilmenge von $\{a,c\}^*$.

Falls das Wort mit c anfängt, wird das erste vorkommende a zum mittleren a beim $\mathcal{L}_2$ zugeordnet. Was im $\mathcal{L}_1$ übrig bleibt, enthalten nur a und c, ist also eine Teilmenge von $\{a,c\}^*$. $\square$

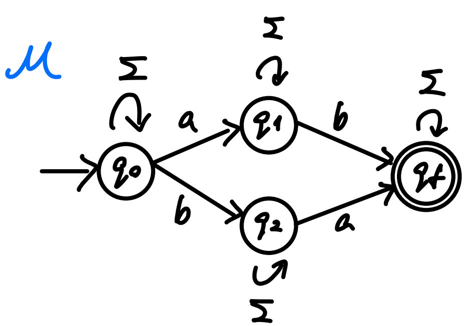
Gilt auch für $\mathcal{L}(\{b,c\}^* b \{b,c\}^*) = \mathcal{L}(\{c\}^* b \{b,c\}^*)$

② Beweis von $\underbrace{\mathcal{L}\left(\begin{matrix} \Sigma^* a \Sigma^* b \Sigma^* \\ \cup\ \Sigma^* b \Sigma^* a \Sigma^* \end{matrix}\right)}_{\mathcal{L}_1} = \underbrace{\mathcal{L}\left(\begin{matrix} \{c\}^* a \{a,c\}^* b \Sigma^* \\ \cup\ \{c\}^* b \{b,c\}^* a \Sigma^* \end{matrix}\right)}_{\mathcal{L}_2}$

NFA (Non-Deterministic Finite Automata)

Determinisierung

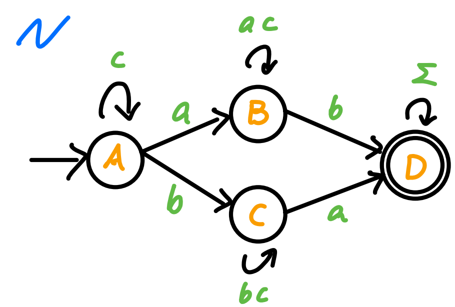
$\mathcal{M}$



	a	b	c
A 0	01	02	0
B 01	01	012	01
C 02	02	012	02
D 012	012	012	012

$\Rightarrow$

$\mathcal{N}$



$\mathcal{L}_1 = \mathcal{L}(\mathcal{M}) \overset{\text{Determinisierung}}{=\!=} \mathcal{L}(\mathcal{N}) = \mathcal{L}_2$ $\square$

Irrelevant!

Nur zum Beweis, dass die beiden Verfahren liefern dasselbe Ergebnis.

Fragen?