Vorlesungsskript

Theoretische Informatik II

Wintersemester 2007/2008

Prof. Dr. Johannes Köbler
Humboldt-Universität zu Berlin
Lehrstuhl Komplexität und Kryptografie

5. November 2007

Inhaltsverzeichnis

Kapitel 1

Reguläre Sprachen

1.1 Endliche Automaten

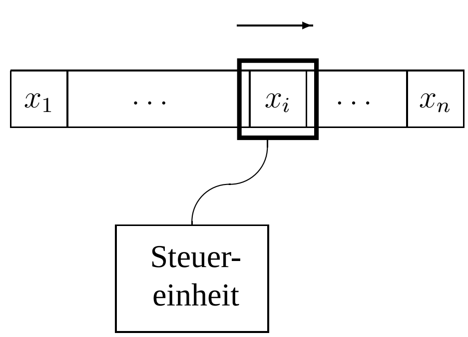
Rechenmaschinen spielen in der Informatik eine zentrale Rolle. Hier beschäftigen wir uns mit mathematischen Modellen für Maschinentypen von unterschiedlicher Berechnungskraft. In der Vorlesung Theoretische Informatik 1 wurde die Turingmaschine als ein universales Berechnungsmodell eingeführt. In dieser Vorlesung werden wir eine Reihe von Einschränkungen dieses Maschinenmodells kennenlernen, die vielfältige praktische Anwendungen haben. Dabei betrachten wir zunächst nur Entscheidungsprobleme, was der Berechnung von $\{0,1\}$-wertigen Funktionen entspricht. Zur Beschreibung der Problemeingaben wird ein Eingabealphabet Σ verwendet.

Definition 1.1 *Ein **Alphabet** ist eine geordnete endliche Menge $\Sigma = \{a_1, \ldots, a_m\}$ von **Zeichen**. Eine Folge $x = x_1 \ldots x_n \in \Sigma^n$ heißt **Wort** (der **Länge** n). Die Menge aller Wörter über Σ ist*

$$\Sigma^* = \bigcup_{n \geq 0} \Sigma^n = \{x_1 \cdots x_n \mid n \geq 0 \textit{ und } x_i \in \Sigma \textit{ für } i = 1, \ldots, n\}.$$

*Das (einzige) Wort der Länge $n = 0$ ist das **leere Wort**, welches wir mit ε bezeichnen.*

Ein endlicher Automat ist eine auf ein Minimum „abgespeckte“ Turingmaschine, die nur kostant viel Speicherplatz zur Verfügung hat und bei Eingaben der Länge n nur n Rechenschritte ausführen darf. Um die gesamte Eingabe lesen zu können, muss der Automat also in jedem Schritt ein Zeichen der Eingabe verarbeiten.



Definition 1.2 *Ein **endlicher Automat** (kurz: DFA; deterministic finite automaton) wird durch ein 5-Tupel $M = (Z, \Sigma, \delta, q_0, E)$ beschrieben, wobei*

- *Z eine endliche Menge von **Zuständen**,*
- *Σ das **Eingabealphabet**,*
- *$\delta : Z \times \Sigma \to Z$ die **Überführungsfunktion**,*
- *$q_0 \in Z$ der **Startzustand** und*
- *$E \subseteq Z$ die Menge der **Endzustände** ist.*

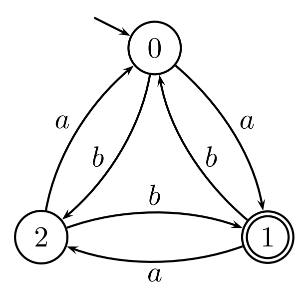
*Die von M **akzeptierte** oder **erkannte Sprache** ist*

$$L(M) = \left\{ x_1 \cdots x_n \in \Sigma^* \,\middle|\, \begin{array}{c} \exists\, q_1, \ldots, q_{n-1} \in Z, q_n \in E: \\ \delta(q_i, x_{i+1}) = q_{i+1} \text{ für } i = 0, \ldots, n-1 \end{array} \right\}$$

Beispiel 1.3 Betrachte den DFA $M = (Z, \Sigma, \delta, 0, E)$ mit $Z = \{0, 1, 2\}$, $\Sigma = \{a, b\}$, $E = \{1\}$ und der Überführungsfunktion

δ	a	b
0	1	2
1	2	0
2	0	1

Graphische Darstellung:



Der Startzustand wird meist durch einen Pfeil und Endzustände werden durch einen doppelten Kreis gekennzeichnet.

Behauptung 1.4 *M akzeptiert die Sprache*

$$L(M) = \{x \in \Sigma^* \mid \#_a(x) - \#_b(x) \equiv 1 \pmod 3\},$$

wobei $\#_a(x)$ die Anzahl der Vorkommen des Buchstabens a in x bezeichnet. (Für $j \equiv k \pmod m$ schreiben wir im Folgenden auch kurz $j \equiv_m k$.)

Beweis Bezeichne $\hat{\delta}(q, x)$ denjenigen Zustand, in dem sich M nach Lesen von x befindet, wenn M im Zustand q gestartet wird. Dann können wir die Funktion

$$\hat{\delta} : Z \times \Sigma^* \to Z$$

induktiv wie folgt definieren. Für $q \in Z$, $x \in \Sigma^*$ und $a \in \Sigma$ sei

$$\begin{aligned} \hat{\delta}(q, \varepsilon) &= q, \\ \hat{\delta}(q, xa) &= \delta(\hat{\delta}(q, x), a). \end{aligned}$$

Da 1 der einzige Endzustand von M ist, reicht es, folgende Kongruenzgleichung zu zeigen:

$$\hat{\delta}(0, x) \equiv_3 \#_a(x) - \#_b(x),$$

Wir beweisen die Kongruenz induktiv über die Länge von x.

Induktionsanfang ($|x| = 0$): klar, da $\hat{\delta}(0, \varepsilon) = 0$ und $\#_a(\varepsilon) = \#_b(\varepsilon) = 0$ ist.

Induktionsschritt ($n \rightsquigarrow n + 1$): Sei $x = x_1 \cdots x_{n+1}$ gegeben. Nach IV ist

$$\hat{\delta}(0, x_1 \cdots x_n) \equiv_3 \#_a(x_1 \cdots x_n) - \#_b(x_1 \cdots x_n).$$

Wegen

$$\delta(i, a) \equiv_3 i + 1 \text{ und } \delta(i, b) \equiv_3 i - 1$$

folgt daher sofort

$$\hat{\delta}(0, x) = \delta(\hat{\delta}(0, x_1 \cdots x_n), x_{n+1}) \equiv_3 \#_a(x) - \#_b(x).$$

■

Eine von einem DFA akzeptierte Sprache wird als **regulär** bezeichnet. Die zugehörige Sprachklasse ist

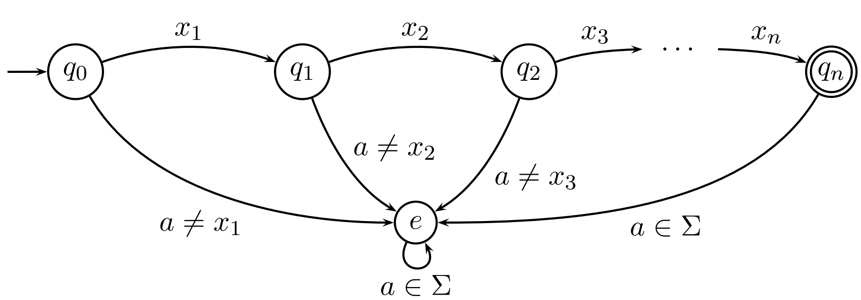
$$\mathsf{REG} = \{L(M) \mid M \text{ ist ein DFA}\}.$$

Um ein intuitives Verständnis für die Berechnungskraft von DFAs zu entwickeln, werden wir uns zunächst intensiv mit der Beantwortung folgender Frage beschäftigen.

Frage: Welche Sprachen gehören zu REG und welche nicht?

Dabei legen wir unseren Überlegungen ein beliebiges aber fest gewähltes Alphabet $\Sigma = \{a_1, \ldots, a_m\}$ zugrunde.

Beobachtung 1.5 *Alle Sprachen, die aus einem einzigen Wort $x = x_1 \cdots x_n \in \Sigma^*$ bestehen, sind regulär. Für folgenden DFA M gilt nämlich $L(M) = \{x\}$.*



Formal lässt sich M also durch das Tupel $M = (Z, \Sigma, \delta, q_0, E)$ mit $Z = \{q_0, \ldots, q_n, e\}$, $E = \{q_n\}$ und der Überführungsfunktion

$$\delta(q, a_j) = \begin{cases} q_{i+1}, & q = q_i \text{ für ein } i \text{ mit } 0 \leq i \leq n-1 \text{ und } a_j = x_{i+1} \\ e, & \text{sonst} \end{cases}$$

beschreiben.

Als nächstes betrachten wir Abschlusseigenschaften der Sprachklasse REG.

Definition 1.6 *Ein* **(k-stelliger) Sprachoperator** *ist eine Abbildung op, die k Sprachen $L_1, \ldots, L_k$ auf eine Sprache $op(L_1, \ldots, L_k)$ abbildet.*

Beispiel 1.7 Der 2-stellige Schnittoperator bildet zwei Sprachen L_1 und L_2 auf die Sprache $L_1 \cap L_2$ ab.

Definition 1.8 *Eine Sprachklasse $\mathcal{K}$ heißt unter op* **abgeschlossen**, *wenn gilt:*

$$L_1, \ldots, L_k \in \mathcal{K} \Rightarrow op(L_1, \ldots, L_k) \in \mathcal{K}.$$

Der **Abschluss** *von $\mathcal{K}$ unter op ist die kleinste Sprachklasse $\mathcal{K}'$, die $\mathcal{K}$ enthält und unter op abgeschlossen ist.*

Beobachtung 1.9 *Mit $L_1, L_2 \in \mathsf{REG}$ sind auch die Sprachen $\overline{L_1} = \Sigma^* \setminus L_1$, $L_1 \cap L_2$ und $L_1 \cup L_2$ regulär. Sind nämlich $M_i = (Z_i, \Sigma, \delta_i, q_0, E_i)$, $i = 1, 2$, DFAs mit $L(M_i) = L_i$, so akzeptiert der DFA*

$$\overline{M_1} = (Z_1, \Sigma, \delta_1, q_0, Z_1 \setminus E_1)$$

das Komplement $\overline{L_1}$ von L_1. Der Schnitt $L_1 \cap L_2$ von L_1 und L_2 wird dagegen von dem DFA

$$M' = (Z_1 \times Z_2, \Sigma, \delta', (q_0, q_0), E_1 \times E_2)$$

mit

$$\delta'((q, p), a) = (\delta_1(q, a), \delta_2(p, a))$$

akzeptiert (M' wird auch **Kreuzproduktautomat** *genannt). Wegen $L_1 \cup L_2 = \overline{(\overline{L_1} \cap \overline{L_2})}$ ist dann aber auch die Vereinigung von L_1 und L_2 regulär. (Wie sieht der zugehörige DFA aus?)*

Aus Beobachtung 1.9 folgt, dass alle endlichen und alle co-endlichen Sprachen regulär sind. Da die in Beispiel 1.3 betrachtete Sprache weder endlich noch co-endlich ist, haben wir damit allerdings noch nicht alle regulären Sprachen erfasst. Es stellt sich die Frage, ob REG neben den mengentheoretischen Operationen Schnitt, Vereinigung und Komplement unter weiteren Operationen wie etwa der **Produktbildung**

$$L_1 L_2 = \{xy \mid x \in L_1, y \in L_2\}$$

(auch **Verkettung** oder **Konkatenation** genannt) oder der Bildung der **Sternhülle**

$$L^* = \bigcup_{n \geq 0} L^n$$

abgeschlossen ist. Die n-fache Potenz L^n von L ist dabei induktiv definiert durch

$$L^0 = \{\varepsilon\},\ L^{n+1} = L^n L.$$

Im übernächsten Abschnitt werden wir sehen, dass die Klasse REG als der Abschluss der endlichen Sprachen unter Vereinigung, Produktbildung und Sternhülle charakterisierbar ist.

Beim Versuch, einen endlichen Automaten für das Produkt $L_1 L_2$ zweier regulärer Sprachen zu konstruieren, stößt man auf die Schwierigkeit, den richtigen Zeitpunkt für den Übergang von (der Simulation von) M_1 zu M_2 zu finden. Unter Verwendung eines nichtdeterministischen Automaten lässt sich dieses Problem jedoch leicht beheben, da dieser den richtigen Zeitpunkt „erraten" kann.

Im nächsten Abschnitt werden wir nachweisen, dass auch nichtdeterministische endliche Automaten nur reguläre Sprachen erkennen können.

1.2 Nichtdeterministische endliche Automaten

Definition 1.10 *Ein **nichtdeterministischer endlicher Automat** (kurz: NFA; nondeterministic finite automaton) $N = (Z, \Sigma, \delta, Q_0, E)$ ist ähnlich aufgebaut wie ein DFA, nur dass er mehrere Startzustände (zusammengefasst in der Menge $Q_0 \subseteq Z$) haben kann und seine Überführungsfunktion*

$$\delta : Z \times \Sigma \to \mathcal{P}(Z)$$

die Potenzmenge $\mathcal{P}(Z)$ von Z als Wertebereich hat. Die von N akzeptierte Sprache ist

$$L(N) \ = \ \left\{ x_1 \cdots x_n \in \Sigma^* \;\middle|\; \begin{array}{l} \exists\, q_0 \in Q_0, q_1, \ldots, q_{n-1} \in Z, q_n \in E\colon \\ q_{i+1} \in \delta(q_i, x_{i+1}) \text{ für } i = 0, \ldots, n-1 \end{array} \right\}$$

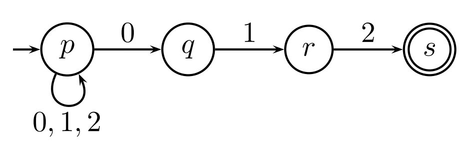
Ein NFA kann also nicht nur eine, sondern mehrere verschiedene Rechnungen ausführen. Die Eingabe gehört bereits dann zu $L(N)$, wenn bei einer dieser Rechnungen nach Lesen des gesamten Eingabewortes ein Endzustand erreicht wird.

Im Gegensatz zu einem DFA, dessen Überführungsfunktion auf der gesamten Menge $Z \times \Sigma$ definiert ist, kann ein NFA „stecken bleiben". Das ist dann der Fall, wenn er in einen Zustand q gelangt, in dem das nächste Eingabezeichen x_i wegen $\delta(q, x_i) = \emptyset$ nicht gelesen werden kann.

Beispiel 1.11 Betrachte den NFA $N = (Z, \Sigma, \delta, Q_0, E)$ mit Zustandsmenge $Z = \{p, q, r, s\}$, Eingabealphabet $\Sigma = \{0, 1, 2\}$, Start- und Endzustandsmenge $Q_0 = \{p\}$ und $E = \{s\}$ sowie der Überführungsfunktion

δ	p	q	r	s
0	$\{p, q\}$	$\emptyset$	$\emptyset$	$\emptyset$
1	$\{p\}$	$\{r\}$	$\emptyset$	$\emptyset$
2	$\{p\}$	$\emptyset$	$\{s\}$	$\emptyset$

Graphische Darstellung:



Offensichtlich akzeptiert N die Sprache $L(N) = \{x012 \mid x \in \Sigma^*\}$ aller Wörter, die mit dem Suffix 012 enden.

Beobachtung 1.12 *Sind $N_i = (Z_i, \Sigma, \delta_i, Q_i, E_i)$ ($i = 1, 2$) NFAs, so werden auch die Sprachen $L(N_1)L(N_2)$ und $L(N_1)^*$ von einem NFA erkannt. Wir können $Z_1 \cap Z_2 = \emptyset$ annehmen. Dann akzeptiert der NFA*

$$N = (Z_1 \cup Z_2, \Sigma, \delta, Q_1, E)$$

mit

$$\delta(p, a) = \begin{cases} \delta_1(p, a), & p \in Z_1 \setminus E_1, \\ \delta_1(p, a) \cup \bigcup_{q \in Q_2} \delta_2(q, a), & p \in E_1, \\ \delta_2(p, a), & \textit{sonst} \end{cases}$$

und

$$E = \begin{cases} E_1 \cup E_2, & Q_2 \cap E_2 \neq \emptyset \\ E_2, & \textit{sonst} \end{cases}$$

die Sprache $L(N_1)L(N_2)$ und der NFA

$$N^* = (Z_1 \cup \{q_{neu}\}, \Sigma, \delta^*, Q_1 \cup \{q_{neu}\}, E_1 \cup \{q_{neu}\})$$

mit

$$\delta^*(p, a) = \begin{cases} \delta(p, a) \cup \bigcup_{q \in Q_1} \delta(q, a), & p \in E_1, \\ \delta(p, a), & \textit{sonst} \end{cases}$$

die Sprache $L(N_1)^*$.

Theorem 1.13 $\mathsf{REG} = \{L(N) \mid N$ *ist ein NFA*$\}$.

Beweis Die Inklusion von links nach rechts ist klar, da jeder DFA auch als NFA aufgefasst werden kann. Für die Gegenrichtung konstruieren wir zu einem NFA $N = (Z, \Sigma, \delta, Q_0, E)$ einen DFA M mit $L(M) = L(N)$. Zunächst erweitern wir die Überführungsfunktion $\delta : Z \times \Sigma \to \mathcal{P}(Z)$ zu $\delta' : \mathcal{P}(Z) \times \Sigma \to \mathcal{P}(Z)$ mittels

$$\delta'(Q, a) = \bigcup_{q \in Q} \delta(q, a).$$

und zeigen folgende Behauptung:

> $\hat{\delta}'(Q_0, x)$ enthält alle von N bei Eingabe x in $|x|$ Schritten erreichbaren Zustände.

Wir beweisen die Behauptung induktiv über die Länge von x.

Induktionsanfang ($|x| = 0$): klar, da $\hat{\delta}'(Q_0, \varepsilon) = Q_0$ ist.

Induktionsschritt ($n-1 \rightsquigarrow n$): Sei $x = x_1 \cdots x_n$ gegeben. Nach Induktionsvoraussetzung enthält

$$Q_{n-1} = \hat{\delta}'(Q_0, x_1 \cdots x_{n-1})$$

alle Zustände, die $N(x)$ in $n-1$ Schritten erreichen kann. Wegen

$$\hat{\delta}'(Q_0, x) = \delta'(Q_{n-1}, x_n) = \bigcup_{q \in Q_{n-1}} \delta(q, x_n)$$

enthält dann aber $\hat{\delta}'(Q_0, x)$ alle Zustände, die $N(x)$ in n Schritten erreichen kann.

Nun ist leicht zu sehen, dass der DFA

$$M = (\mathcal{P}(Z), \Sigma, \delta', Q_0, E')$$

mit

$$\delta'(Q, a) = \bigcup_{q \in Q} \delta(q, a)$$

und

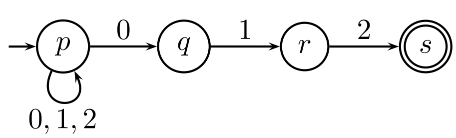
$$E' = \{Q \subseteq Z \mid Q \cap E \neq \emptyset\}$$

äquivalent zu N ist, da für alle Wörter $x \in \Sigma^*$ gilt:

$$\begin{aligned} x \in L(N) &\Leftrightarrow N(x) \text{ kann in genau } |x| \text{ Schritten einen Endzustand erreichen} \\ &\Leftrightarrow \hat{\delta}'(Q_0, x) \cap E \neq \emptyset \\ &\Leftrightarrow \hat{\delta}'(Q_0, x) \in E' \\ &\Leftrightarrow x \in L(M). \end{aligned}$$

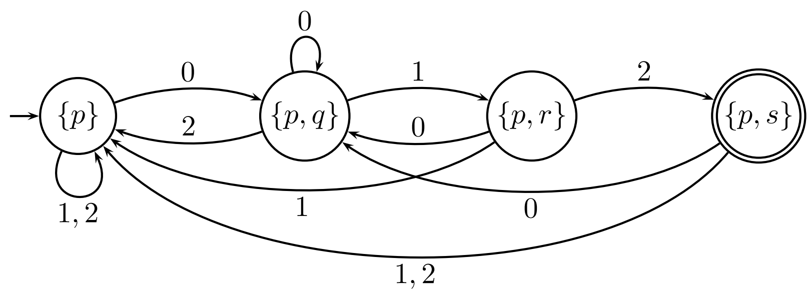
■

Beispiel 1.14 Für den NFA $N = (Z, \Sigma, \delta, Q_0, E)$ aus Beispiel 1.11



ergibt die Konstruktion des vorigen Satzes den folgenden DFA M (nach Entfernung aller vom Startzustand $Q_0 = \{p\}$ aus nicht erreichbaren Zustände):

δ'	0	1	2
$Q_0 = \{p\}$	$\{p, q\}$	$\{p\}$	$\{p\}$
$Q_1 = \{p, q\}$	$\{p, q\}$	$\{p, r\}$	$\{p\}$
$Q_2 = \{p, r\}$	$\{p, q\}$	$\{p\}$	$\{p, s\}$
$Q_3 = \{p, s\}$	$\{p, q\}$	$\{p\}$	$\{p\}$



Im obigen Beispiel wurden für die Konstruktion des DFA M aus dem NFA N nur 4 der insgesamt $2^{\|Z\|} = 16$ Zustände benötigt, da die übrigen 12 Zustände in $\mathcal{P}(Z)$ nicht vom Startzustand $Q_0 = \{p\}$ aus erreichbar sind. Es gibt jedoch Beispiele, bei denen alle $2^{\|Z\|}$ Zustände in $\mathcal{P}(Z)$ für die Konstruktion des so genannten **Potenzmengenautomaten** M benötigt werden (siehe Übungen).

1.3 Reguläre Ausdrücke

Wir haben uns im letzten Abschnitt davon überzeugt, dass auch NFAs nur reguläre Sprachen erkennen können:

$$\mathsf{REG} = \{L(M) \mid M \text{ ist ein DFA}\} = \{L(N) \mid N \text{ ist ein NFA}\}.$$

In diesem Abschnitt werden wir eine weitere Charakterisierung der regulären Sprachen kennen lernen:

> REG ist die Klasse aller Sprachen, die sich mittels der Operationen Vereinigung, Durchschnitt, Komplement, Produkt und Sternhülle aus der leeren Menge und den einelementigen Sprachen bilden lassen.

Tatsächlich kann hierbei sogar auf die Durchschnitts- und Komplementbildung verzichtet werden.

Definition 1.15 *Die Menge der **regulären Ausdrücke** γ (über einem Alphabet Σ) und die durch γ dargestellte Sprache $L(\gamma)$ sind induktiv wie folgt definiert. Die Symbole $\emptyset$, ϵ und a ($a \in \Sigma$) sind reguläre Ausdrücke, die*

- *die leere Sprache $L(\emptyset) = \emptyset$,*
- *die Sprache $L(\epsilon) = \{\varepsilon\}$ und*
- *für jedes Zeichen $a \in \Sigma$ die Sprache $L(a) = \{a\}$*

beschreiben. Sind α und β reguläre Ausdrücke, die die Sprachen $L(\alpha)$ und $L(\beta)$ beschreiben, so sind auch $\alpha\beta$, $(\alpha|\beta)$ und $(\alpha)^*$ reguläre Ausdrücke, die die Sprachen

- $L(\alpha\beta) = L(\alpha)L(\beta)$,
- $L(\alpha|\beta) = L(\alpha) \cup L(\beta)$ *und*
- $L((\alpha)^*) = L(\alpha)^*$

beschreiben.

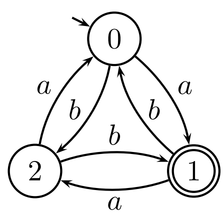
Beispiel 1.16 Über $\Sigma = \{0, 1\}$ sind ϵ^*, $\emptyset^*$, $(0|1)^*00$ und $(\epsilon 0|\emptyset 1^*)$ reguläre Ausdrücke, die folgende Sprachen beschreiben:

γ	ϵ^*	$\emptyset^*$	$(0\|1)^*00$	$(\epsilon 0\|\emptyset 1^*)$
$L(\gamma)$	$\{\varepsilon\}^* = \{\varepsilon\}$	$\emptyset^* = \{\varepsilon\}$	$\{x00 \mid x \in \Sigma^*\}$	$\{0\}$

Bemerkung 1.17

- *Um Klammern zu sparen, definieren wir folgende* ***Präzedenzordnung****:*
  Der Sternoperator * *bindet stärker als der Produktoperater und dieser wiederum stärker als der Vereinigungsoperator* $|$.
- *Da der reguläre Ausdruck* $\gamma\gamma^*$ *die Sprache* $L(\gamma)^+$ *beschreibt, verwenden wir* γ^+ *als Abkürzung für den Ausdruck* $\gamma\gamma^*$.

Beispiel 1.18 Betrachte den DFA M:



Um für die von M erkannte Sprache

$$L(M) = \{x \in \{a, b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}$$

einen regulären Ausdruck zu finden, betrachten wir zunächst die Sprache $L_1 = \{x \in \{a, b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 0\}$. Da sich L_1 durch den regulären Ausdruck

$$\gamma = (a(ab)^*(aa|b) \mid b(ba)^*(a|bb))^*$$

beschreiben lässt, erhalten wir für $L(M)$ den regulären Ausdruck $\gamma(ba)^*(a|bb)$.

Theorem 1.19 $\mathsf{REG} = \{L(\gamma) \mid \gamma$ *ist ein regulärer Ausdruck*$\}$.

Beweis Die Inklusion von rechts nach links ist klar, da die Basisausdrücke $\emptyset$, ϵ und a, $a \in \Sigma^*$, nur reguläre Sprachen beschreiben und die Sprachklasse REG unter Produkt, Vereinigung und Sternhülle abgeschlossen ist (siehe Beobachtungen 1.9 und 1.12).

Für die Gegenrichtung konstruieren wir zu einem DFA M einen regulären Ausdruck γ mit $L(\gamma) = L(M)$. Sei also $M = (Z, \Sigma, \delta, q_0, E)$ ein DFA, wobei wir annehmen können, dass $Z = \{1, \ldots, m\}$ und $q_0 = 1$ ist. Dann lässt sich $L(M)$ als Vereinigung

$$L(M) = \bigcup_{q \in E} L_{1,q}$$

von Sprachen der Form

$$L_{p,q} = \{x \in \Sigma^* \mid \hat{\delta}(p, x) = q\}$$

darstellen. Folglich reicht es zu zeigen, dass die Sprachen $L_{p,q}$ durch reguläre Ausdrücke beschreibbar sind. Hierzu betrachten wir die Sprachen

$$L^r_{p,q} = \{x \in \Sigma^* \mid \hat{\delta}(p, x) = q \text{ und für } i = 1, \ldots, n-1 \text{ gilt } \hat{\delta}(p, x_1 \cdots x_i) \le r\}.$$

Wegen $L_{p,q} = L^m_{p,q}$ reicht es, reguläre Ausdrücke $\gamma^r_{p,q}$ für die Sprachen $L^r_{p,q}$ anzugeben. Im Fall $r = 0$ enthält

$$L^0_{p,q} = \begin{cases} \{a \in \Sigma \mid \delta(p, a) = q\} \cup \{\varepsilon\}, & p = q, \\ \{a \in \Sigma \mid \delta(p, a) = q\}, & \text{sonst} \end{cases}$$

nur Buchstaben (und eventuell das leere Wort) und ist somit leicht durch einen regulären Ausdruck $\gamma^0_{p,q}$ beschreibbar. Wegen

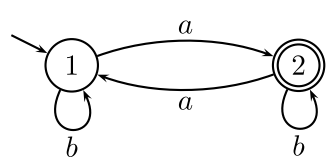
$$L^{r+1}_{p,q} \quad = \quad L^r_{p,q} \cup L^r_{p,r+1}(L^r_{r+1,r+1})^* L^r_{r+1,q}$$

lassen sich aus den regulären Ausdrücken $\gamma^r_{p,q}$ für die Sprachen $L^r_{p,q}$ leicht reguläre Ausdrücke für die Sprachen $L^{r+1}_{p,q}$ gewinnen:

$$\gamma^{r+1}_{p,q} = \gamma^r_{p,q} | \gamma^r_{p,r+1} (\gamma^r_{r+1,r+1})^* \gamma^r_{r+1,q}.$$

■

Beispiel 1.20 Betrachte den DFA



Da M insgesamt $m = 2$ Zustände und nur den Endzustand 2 besitzt, ist

$$L(M) = \bigcup_{q \in E} L_{1,q} = L_{1,2} = L^2_{1,2} = L(\gamma^2_{1,2}).$$

Um $\gamma_{1,2}^2$ zu berechnen, benutzen wir die Rekursionsformel

$$\gamma_{p,q}^{r+1} = \gamma_{p,q}^r | \gamma_{p,r+1}^r (\gamma_{r+1,r+1}^r)^* \gamma_{r+1,q}^r$$

und erhalten

$$\begin{aligned}
\gamma_{1,2}^2 &= \gamma_{1,2}^1 | \gamma_{1,2}^1 (\gamma_{2,2}^1)^* \gamma_{2,2}^1, \\
\gamma_{1,2}^1 &= \gamma_{1,2}^0 | \gamma_{1,1}^0 (\gamma_{1,1}^0)^* \gamma_{1,2}^0, \\
\gamma_{2,2}^1 &= \gamma_{2,2}^0 | \gamma_{2,1}^0 (\gamma_{1,1}^0)^* \gamma_{1,2}^0.
\end{aligned}$$

Um einen regulären Ausdruck für $L(M) = L(b^*a(b|ab^*a)^*)$ zu erhalten, genügt es also, die regulären Ausdrücke $\gamma_{1,1}^0$, $\gamma_{1,2}^0$, $\gamma_{2,1}^0$, $\gamma_{2,2}^0$, $\gamma_{1,2}^1$, $\gamma_{2,2}^1$ und $\gamma_{1,2}^2$ zu berechnen:

r	p,q			
	$1,1$	$1,2$	$2,1$	$2,2$
0	$\epsilon\|b$	a	a	$\epsilon\|b$
1	-	$\underbrace{a\|(\epsilon\|b)(\epsilon\|b)^*a}_{b^*a}$	-	$\underbrace{(\epsilon\|b)\|a(\epsilon\|b)^*a}_{\epsilon\|b\|ab^*a}$
2	-	$\underbrace{b^*a\|b^*a(\epsilon\|b\|ab^*a)^*(\epsilon\|b\|ab^*a)}_{b^*a(b\|ab^*a)^*}$	-	-

1.4 Relationalstrukturen

Sei A eine nichtleere Menge, R_i eine k_i-stellige Relation auf A, d.h. $R_i \subseteq A^{k_i}$ für $i = 1, \ldots, n$. Dann heißt $(A; R_1, \ldots, R_n)$ ***Relationalstruktur***. Die Menge A heißt ***Grundmenge***, ***Trägermenge*** oder ***Individuenbereich*** der Relationalstruktur.

Bemerkung 1.21

- *Wir werden hier hauptsächlich den Fall $n = 1$, $k_1 = 2$, also (A, R) mit $R \subseteq A \times A$ betrachten. Man nennt dann R eine* ***(binäre) Relation*** *auf A.*
- *Oft wird für $(a, b) \in R$ auch die* ***Infix-Schreibweise*** *aRb benutzt.*

Beispiel 1.22

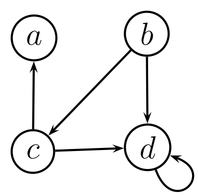
- (F, M) mit $F := \{f \mid f$ ist Fluss in Europa$\}$ und
  $M := \{(f, g) \in F \times F \mid f$ mündet in $g\}$.
- (U, B) mit $U := \{x \mid x$ ist Berliner $\}$ und
  $B := \{(x, y) \in U \times U \mid x$ ist Bruder von $y\}$.

- $(\mathsf{P}(M), \subseteq)$, wobei $\mathsf{P}(M)$ die Potenzmenge einer beliebigen Menge M und $\subseteq$ die Inklusionsbeziehung auf den Teilmengen von M ist.
- (A, Id_A), wobei $Id_A = \{(x,x) \mid x \in A\}$ die ***Identität auf*** A ist.
- $(\mathbb{R}, \leq)$.
- $(\mathbb{Z}, |)$, wobei $|$ die "teilt"-Relation bezeichnet.
- $(\mathcal{F}ml, \Rightarrow)$ mit $\mathcal{F}ml := \{F \mid F$ ist aussagenlogische Formel$\}$ und $\Rightarrow = \{(F,G) \in \mathcal{F}ml \times \mathcal{F}ml \mid G$ ist aussagenlogische Folgerung von $F\}$.

Graphische Darstellung von Relationen

Eine Relation R auf einer endlichen Menge A kann durch einen ***gerichteten Graphen*** $G = (V,E)$ mit ***Knotenmenge*** $V = A$ und ***Kantenmenge*** $E = R$ veranschaulicht werden. Hierzu stellen wir jedes Element $x \in A$ als einen Knoten dar und verbinden jedes Knotenpaar $(x,y) \in R$ durch eine gerichtete Kante (Pfeil). Zwei durch eine Kante verbundene Knoten heißen ***benachbart*** oder ***adjazent***.

Beispiel 1.23 Für die Relation (A,R) mit $A = \{a,b,c,d\}$ und $R = \{(b,c),(b,d),(c,a),(c,d),(d,d)\}$ erhalten wir folgende graphische Darstellung.



Der ***Ausgangsgrad*** eines Knotens $x \in V$ ist $d_{out}(x) = \|R(x)\|$, wobei $R(x) = \{y \in V \mid xRy\}$ der ***Nachbereich*** von x ist. Entsprechend ist $d_{in}(x) = \|\{y \in V \mid yRx\}\|$ der ***Eingangsgrad*** von x. Falls R symmetrisch ist, können die Pfeilspitzen auch weggelassen werden. In diesem Fall ist $d(x) = d_{in}(x) = d_{out}(x)$ der ***Grad*** von x. Ist R zudem irreflexiv, so erhalten wir einen (schleifenfreien) ***Graphen***.

Darstellung durch eine Adjazenzmatrix

Eine Relation R auf einer endlichen (geordneten) Menge $A = \{a_1, \ldots, a_n\}$ lässt sich durch eine boolesche $n \times n$-Matrix $M_R = (m_{ij})$ mit

$$m_{ij} := \begin{cases} 1, & a_i R a_j, \\ 0, & \text{sonst} \end{cases}$$

darstellen. Beispielsweise hat die Relation

$$R = \{(b,c),(b,d),(c,a),(c,d),(d,d)\}$$

auf der Menge $A = \{a, b, c, d\}$ die Matrixdarstellung

$$M_R = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Darstellung durch eine Adjazenzliste

Eine weitere Möglichkeit besteht darin, eine endliche Relation R in Form einer Tabelle darzustellen, die jedem Element $x \in A$ seinen Nachbereich $R(x)$ in Form einer Liste zuordnet:

x	$R(x)$
a	-
b	c, d
c	a, d
d	d

1.4.1 Eigenschaften von Relationen

Sei R eine Relation auf A. Dann heißt R

reflexiv,	falls $\forall x \in A : xRx$ (d.h. $Id_A \subseteq R$),
irreflexiv,	falls $\forall x \in A : \neg xRx$ (d.h. $Id_A \subseteq \overline{R}$),
symmetrisch,	falls $\forall x, y \in A : xRy \Rightarrow yRx$ (d.h. $R \subseteq R^T$),
asymmetrisch,	falls $\forall x, y \in A : xRy \Rightarrow \neg yRx$ (d.h. $R \subseteq \overline{R^T}$),
antisymmetrisch,	falls $\forall x, y \in A : xRy \wedge yRx \Rightarrow x = y$ (d.h. $R \cap R^T \subseteq Id$),
konnex,	falls $\forall x, y \in A : xRy \vee yRx$ (d.h. $A \times A \subseteq R \cup R^T$),
semikonnex,	falls $\forall x, y \in A : x \neq y \Rightarrow xRy \vee yRx$ (d.h. $\overline{Id} \subseteq R \cup R^T$),
transitiv,	falls $\forall x, y, z \in A : xRy \wedge yRz \Rightarrow xRz$ (d.h. $R^2 \subseteq R$)

gilt.

Beispiel 1.24

- Die Relation ”ist Schwester von” ist zwar in einer reinen Damengesellschaft symmetrisch, i.a. jedoch weder symmetrisch noch asymmetrisch noch antisymmetrisch.
- $(\mathbb{R}, <)$ ist irreflexiv, asymmetrisch, transitiv und semikonnex.
- $(\mathbb{R}, \leq)$ und $(\mathsf{P}(M), \subseteq)$ sind reflexiv, antisymmetrisch und transitiv.
- $(\mathbb{R}, \leq)$ ist auch konnex und $(\mathsf{P}(M), \subseteq)$ ist im Fall $\|M\| \leq 1$ zwar auch konnex, aber im Fall $\|M\| \geq 2$ weder semikonnex noch konnex.

Operationen auf Relationen

Da Relationen Mengen sind, sind auf ihnen die mengentheoretischen Operationen ***Durchschnitt***, ***Vereinigung***, ***Komplement*** und ***Differenz*** definiert. Seien R und S Relationen auf A, dann ist

$$\begin{array}{lcl} R \cap S &=& \{(x,y) \in A \times A \mid xRy \wedge xSy\}, \\ R \cup S &=& \{(x,y) \in A \times A \mid xRy \vee xSy\}, \\ R - S &=& \{(x,y) \in A \times A \mid xRy \wedge \neg xSy\}, \\ \overline{R} &=& (A \times A) - R. \end{array}$$

Sei allgemeiner $\mathcal{M} \subseteq \mathcal{P}(A \times A)$ eine beliebige Menge von Relationen auf A. Dann sind der **Schnitt über** $\mathcal{M}$ und die **Vereinigung über** $\mathcal{M}$ folgende Relationen:

$$\begin{array}{lcl} \bigcap \mathcal{M} &=& \{(x,y) \mid \forall R \in \mathcal{M} : xRy\} \\ \bigcup \mathcal{M} &=& \{(x,y) \mid \exists R \in \mathcal{M} : xRy\} \end{array}$$

Weiterhin ist die **Inklusionsrelation** $R \subseteq S$ auf Relationen definiert:

$$R \subseteq S \Leftrightarrow \forall x, y : xRy \rightarrow xSy.$$

Die ***transponierte (konverse) Relation*** zu R ist

$$R^T := \{(y,x) \mid xRy\}.$$

R^T wird oft auch mit R^{-1} bezeichnet. Zum Beispiel ist $(\mathbb{R}, \leq^T) = (\mathbb{R}, \geq)$.
Eine wichtige zweistellige Operation auf der Menge $\mathsf{P}(A \times A)$ aller Relationen auf A ist das Relationenprodukt (auch Komposition genannt).
Das ***Produkt*** zweier Relationen R und S auf A ist

$$R \circ S := \{(x,z) \mid \exists y : xRy \wedge ySz\}.$$

Übliche Bezeichnungen für das Relationenprodukt sind auch $R\,;\!S$ und $R \cdot S$ oder einfach RS. Für $\underbrace{R \circ \cdots \circ R}_{\text{n-mal}}$ wird auch R^n geschrieben. Dabei ist $R^0 = Id$.

Vorsicht: Das n-fache Relationenprodukt von R sollte nicht mit dem n-fachen kartesischen Produkt der *Menge* R verwechselt werden. Wir vereinbaren, dass R^n das n-fache Relationenprodukt bezeichnen soll, falls R eine Relation ist.

Beispiel 1.25 Ist B die Relation ”ist Bruder von”, V ”ist Vater von”, M ”ist Mutter von” und $E = V \cup M$ ”ist Elternteil von”, so ist $B \circ E$ die Relation ”ist Onkel von”.

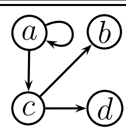
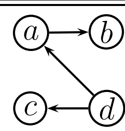
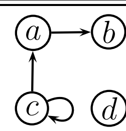
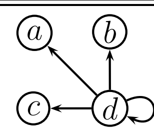
Sind $M_R = (r_{ij})$ und $M_S = (s_{ij})$ boolesche $n \times n$-Matrizen für R und S, so erhalten wir für $T = R \circ S$ die Matrix $M_T = (t_{ij})$ mit

$$t_{ij} := \bigvee_{k=1,\dots,n} (r_{ik} \wedge s_{kj})$$

Der Nachbereich $T(x)$ von x bzgl. der Relation $T = R \circ S$ berechnet sich zu

$$T(x) = \bigcup\{S(y) \mid y \in R(x)\} = \bigcup_{y \in R(x)} S(y).$$

Beispiel 1.26 Betrachte die Relationen $R = \{(a,a),(a,c),(c,b),(c,d)\}$ und $S = \{(a,b),(d,a),(d,c)\}$ auf der Menge $A = \{a,b,c,d\}$.

Relation	R	S	$R \circ S$	$S \circ R$
Graph				
Adjazenz- matrix	$\begin{array}{cccc}1&0&1&0\\0&0&0&0\\0&1&0&1\\0&0&0&0\end{array}$	$\begin{array}{cccc}0&1&0&0\\0&0&0&0\\0&0&0&0\\1&0&1&0\end{array}$	$\begin{array}{cccc}0&1&0&0\\0&0&0&0\\1&0&1&0\\0&0&0&0\end{array}$	$\begin{array}{cccc}0&0&0&0\\0&0&0&0\\0&0&0&0\\1&1&1&1\end{array}$
Adjazenz- liste	$a : a, c$ $b :$ - $c : b, d$ $d :$ -	$a : b$ $b :$ - $c :$ - $d : a, c$	$a : b$ $b :$ - $c : a, c$ $d :$ -	$a :$ - $b :$ - $c :$ - $d : a, b, c, d$

Beobachtung: Das Beispiel zeigt, dass das Relationenprodukt nicht kommutativ ist, d.h. i.a. gilt nicht $R \circ S = S \circ R$.

Als nächstes zeigen wir, dass die Menge $\mathcal{R} = \mathcal{P}(A \times A)$ aller binären Relationen auf A mit dem Relationenprodukt $\circ$ als binärer Operation und der Relation Id_A als neutralem Element eine Halbgruppe (oder ***Monoid***) bildet.

Theorem 1.27 *Seien Q, R, S Relationen auf A. Dann gilt*

(i) $(Q \circ R) \circ S = Q \circ (R \circ S)$, *d.h. $\circ$ ist assoziativ,*

(ii) $Id \circ R = R \circ Id = R$, *d.h. Id ist neutrales Element.*

Beweis

(i) Es gilt:

$$\begin{array}{rcl} x\,(Q \circ R) \circ S\; y & \Leftrightarrow & \exists u \in A : x\,(Q \circ R)\; u \;\wedge\; u\; S\; y \\ & \Leftrightarrow & \exists u \in A : (\exists v \in A : x\; Q\; v\; R\; u) \;\wedge\; u\; S\; y \\ & \Leftrightarrow & \exists u, v \in A : x\; Q\; v\; R\; u\; S\; y \\ & \Leftrightarrow & \exists v \in A : x\; Q\; v \;\wedge\; (\exists u \in A : v\; R\; u \;\wedge\; u\; S\; y) \\ & \Leftrightarrow & \exists v \in A : x\; Q\; v\; (R \circ S)\; y \\ & \Leftrightarrow & x\; Q \circ (R \circ S)\; y \end{array}$$

(ii) Wegen $x\ Id \circ R\ y \Leftrightarrow \exists z : x = z \ \wedge \ z\ R\ y \Leftrightarrow x\ R\ y$ folgt $Id \circ R = R$. Die Gleichheit $R \circ Id = R$ folgt analog.

■

Manchmal steht man vor der Aufgabe, eine gegebene Relation R durch eine möglichst kleine Modifikation in eine Relation R' mit vorgegebenen Eigenschaften zu überführen. Will man dabei alle in R enthaltenen Paare beibehalten, dann sollte R' aus R durch Hinzufügen möglichst weniger Paare hervorgehen.

Es lässt sich leicht nachprüfen, dass der Schnitt über eine Menge reflexiver (bzw. transitiver oder symmetrischer) Relationen wieder reflexiv (bzw. transitiv oder symmetrisch) ist. Folglich existiert zu jeder Relation R auf einer Menge A eine kleinste reflexive (bzw. transitive oder symmetrische) Relation R', die R enthält.

Definition 1.28 *Sei R eine Relation.*

- *Die* ***reflexive Hülle*** *von R ist*
$$h_{\mathit{refl}}(R) = \bigcap \{S \subseteq A \times A \mid S \text{ ist reflexiv und } R \subseteq S\}.$$
- *Die* ***symmetrische Hülle*** *von R ist*
$$h_{\mathit{sym}}(R) = \bigcap \{S \subseteq A \times A \mid S \text{ ist symmetrisch und } R \subseteq S\}.$$
- *Die* ***transitive Hülle*** *von R ist*
$$R^+ = \bigcap \{S \subseteq A \times A \mid S \text{ ist transitiv und } R \subseteq S\}.$$
- *Die* ***reflexiv-transitive Hülle*** *von R ist*
$$R^* = \bigcap \{S \subseteq A \times A \mid S \text{ ist reflexiv, transitiv und } R \subseteq S\}.$$

Theorem 1.29 *Sei R eine Relation auf A.*

(i) $h_{\mathit{refl}}(R) = R \cup Id_A$,

(ii) $h_{\mathit{sym}}(R) = R \cup R^T$,

(iii) $R^+ = \bigcup_{n \geq 1} R^n$,

(iv) $R^* = \bigcup_{n \geq 0} R^n$.

Beweis Siehe Übungen. ■

Anschaulich besagt der vorhergehende Satz, dass ein Paar (a, b) genau dann in der reflexiv-transitiven Hülle R^* von R ist, wenn es ein $n \geq 0$ gibt mit aR^nb, d.h. es gibt Elemente $x_0, \ldots, x_n \in A$ mit $x_0 = a$, $x_n = b$ und

$$x_0Rx_1Rx_2 \cdots x_{n-1}Rx_n.$$

In der Graphentheorie nennt man $x_0 \cdots x_n$ einen ***Weg der Länge*** n ***von*** a ***nach*** b.

1.4.2 Äquivalenz- und Ordnungsrelationen

Die nachfolgende Tabelle gibt einen Überblick über die definierenden Eigenschaften der wichtigsten Relationalstrukturen.

	refl.	sym.	trans.	asym.	antisym.	konnex	semikon.
Äquivalenzrel.	✓	✓	✓				
(Halb-)Ordnung	✓		✓		✓		
Striktordnung			✓	✓			
lineare Ord.			✓		✓	✓	
lin. Striktord.			✓	✓			✓
schwache Ord.			✓			✓	
Quasiordnung	✓		✓				

In der Tabelle sind nur die definierenden Eigenschaften durch ein ”✓” gekennzeichnet. Das schließt nicht aus, dass gleichzeitig auch noch weitere Eigenschaften vorliegen können.

Wir betrachten zunächst eine Reihe von Beispielen für ***Äquivalenzrelationen***, die durch die drei Eigenschaften reflexiv, symmetrisch und transitiv definiert sind.

Beispiel 1.30

- Auf der Menge aller Geraden im $\mathbb{R}^2$ die Parallelität.
- Auf der Menge aller Menschen ”im gleichen Jahr geboren wie”.
- Auf $\mathbb{Z}$ die Relation ”gleicher Rest bei Division durch m”.
- Auf der Menge aller aussagenlogischen Formeln die semantische Äquivalenz.

Ist E eine Äquivalenzrelation, so nennt man den zu x gehörigen Nachbereich $E(x)$ die ***von*** x ***repräsentierte Äquivalenzklasse*** und bezeichnet sie mit $[x]_E$ oder einfach mit $[x]$. Die durch E auf A induzierte Partition $\{[x] \mid x \in A\}$ wird ***Quotienten- oder Faktormenge*** genannt und mit A/E bezeichnet. Die Anzahl der Äquivalenzklassen von E wird auch als der ***Index*** von E bezeichnet.

Definition 1.31 *Eine Familie $\{M_i \mid i \in I\}$ von nichtleeren Teilmengen $M_i \subseteq A$ heißt **Partition** der Menge A, falls gilt:*

*a) die Mengen M_i **überdecken** A, d.h. $A = \bigcup_{i \in I} M_i$ und*

*b) die Mengen M_i sind **paarweise disjunkt**, d.h. für je zwei verschiedene Mengen $M_i \neq M_j$ gilt $M_i \cap M_j = \emptyset$.*

Wie der nächste Satz zeigt, beschreiben Äquivalenzrelationen auf A und Partitionen von A den selben Sachverhalt.

Theorem 1.32 *Sei E eine Relation auf A. Dann sind folgende Aussagen äquivalent.*

(i) *E ist eine Äquivalenzrelation auf A.*

(ii) *Für alle $x, y \in A$ gilt*

$$xEy \Leftrightarrow E(x) = E(y) \qquad (*)$$

(iii) *E ist reflexiv und $\{E(x) \mid x \in A\}$ ist eine Partition von A.*

Beweis

$(i) \Rightarrow (ii)$ Sei E eine Äquivalenzrelation auf A. Da E transitiv ist, impliziert xEy die Inklusion $E(y) \subseteq E(x)$:

$$z \in E(y) \Rightarrow yEz \Rightarrow xEz \Rightarrow z \in E(x).$$

Da E symmetrisch ist, folgt aus xEy aber auch $E(x) \subseteq E(y)$.

Umgekehrt folgt aus $E(x) = E(y)$ wegen der Reflexivität von E, dass $x \in E(x) = E(y)$ enthalten ist, und somit xEy. Dies zeigt, dass E die Äquivalenz $(*)$ erfüllt.

$(ii) \Rightarrow (iii)$ Falls E die Bedingung $(*)$ erfüllt, so folgt sofort xEx (wegen $E(x) = E(x)$) und folglich überdecken die Nachbereiche $E(x)$ (wegen $x \in E(x)$) die Menge A.

Ist $E(x) \cap E(y) \neq \emptyset$ und z ein Element in $E(x) \cap E(y)$, so gilt xEz und yEz und daher folgt $E(x) = E(z) = E(y)$. Da also je zwei Nachbereiche $E(x)$ und $E(y)$ entweder gleich oder disjunkt sind, bildet $\{E(x) \mid x \in A\}$ sogar eine Partition von A.

$(iii) \Rightarrow (i)$ Wird schließlich A von den Mengen $E(x)$ partitioniert, wobei $x \in E(x)$ für alle $x \in A$ gilt, so folgt

$$xEy \Leftrightarrow y \in E(x) \cap E(y) \Leftrightarrow E(x) = E(y).$$

Daher übertragen sich die Eigenschaften Reflexivität, Symmetrie und Transitivität unmittelbar von der Gleichheitsrelation auf E.

$\blacksquare$

Beispiel 1.33 Für die weiter oben betrachteten Äquivalenzrelationen erhalten wir folgende Klasseneinteilungen.

- Für die Parallelität auf der Menge aller Geraden im $\mathbb{R}^2$: alle Geraden mit derselben Richtung (oder Steigung) bilden jeweils eine Äquivalenzklasse.

- Für die Relation "im gleichen Jahr geboren wie" auf der Menge aller Menschen: jeder Jahrgang bildet eine Äquivalenzklasse.

- Für die Relation "gleicher Rest bei Division durch m" auf $\mathbb{Z}$: jede der m Restklassen $[0], [1], \ldots, [m-1]$ mit

  $$[r] = \{a \in \mathbb{Z} \mid a \bmod m = r\}$$

  bildet eine Äquivalenzklasse.

Die kleinste Äquivalenzrelation auf A ist die **Identität** Id_A, die größte die **Allrelation** $A \times A$. Die Äquivalenzklassen der Identität enthalten jeweils nur ein Element, d.h. $A/Id_A = \{\{x\} \mid x \in A\}$, und die Allrelation erzeugt nur eine Äquivalenzklasse, nämlich $A/(A \times A) = \{A\}$.

Für zwei Äquivalenzrelationen $E \subseteq E'$ sind auch die Äquivalenzklassen $[x]_E$ von E in den Klassen $[x]_{E'}$ von E' enthalten. Folglich ist jede Äquivalenzklasse von E' die Vereinigung von (evtl. mehreren) Äquivalenzklassen von E_1. Im Fall $E \subseteq E'$ sagt man auch, E_1 bewirkt eine **feinere** Partitionierung als E_2. Demnach ist die Identität die **feinste** und die Allrelation die **gröbste** Äquivalenzrelation.

Da der Schnitt über eine Menge von Äquivalenzrelationen wieder eine Äquivalenzrelation ist, können wir für eine beliebige Relation R auf einer Menge A die kleinste R umfassende Äquivalenzrelation definieren:

$$h_{\text{äq}}(R) := \bigcap \{E \mid E \text{ ist eine Äquivalenzrelation auf } A \text{ mit } R \subseteq E\}$$

In der Sprache der Graphentheorie werden die durch $h_{\text{äq}}(R)$ generierten Äquivalenzklassen auch die **schwachen Zusammenhangskomponenten** des gerichteten Graphen (A, R) genannt (siehe Übungen).

Als nächstes betrachten wir Ordnungen.

Definition 1.34 *(A, R) heißt **Ordnung** (auch **Halbordnung** oder **partielle Ordnung**), wenn R eine reflexive, antisymmetrische und transitive Relation auf A ist.*

Beispiel 1.35

- $(\mathbb{Z}, \leq)$ und $(\mathbb{N}, |)$ sind Ordnungen. Erstere ist linear, letztere nicht.

- Für jede Menge M ist die relationale Struktur $(\mathcal{P}(M); \subseteq)$ eine Ordnung. Diese ist nur im Fall $\|M\| \leq 1$ linear.

- Auf der Menge $\mathcal{A}(M)$ aller Äquivalenzrelationen auf M die Relation "feiner als". Dabei ist, wie wir gesehen haben, E_1 eine Verfeinerung von E_2, falls E_1 in E_2 enthalten ist. In diesem Fall bewirkt E_1 nämlich eine feinere Klasseneinteilung auf M als E_2, da jede Äquivalenzklasse von E_1 in einer Äquivalenzklasse von E_2 enthalten ist.

- Ist R eine Ordnung auf A und $B \subseteq A$, so heißt die Ordnung $R_B = R \cap (B \times B)$ die ***Einschränkung*** (oder ***Restriktion***) von R auf B. Beispielsweise ist $(\mathcal{A}(M); \subseteq)$ die Einschränkung von $(\mathcal{P}(M \times M); \subseteq)$ auf $\mathcal{A}(M)$.

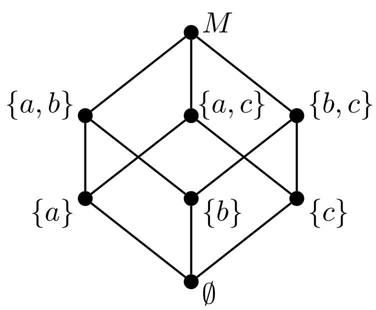
Ordnungen lassen sich sehr anschaulich durch Hasse-Diagramme darstellen. Sei $\leq$ eine Ordnung auf A und sei $<$ die Relation $\leq \cap \overline{Id_A}$. Um die Ordnung $\leq$ in einem ***Hasse-Diagramm*** darzustellen, wird nur der Graph der ***Nachbarrelation***

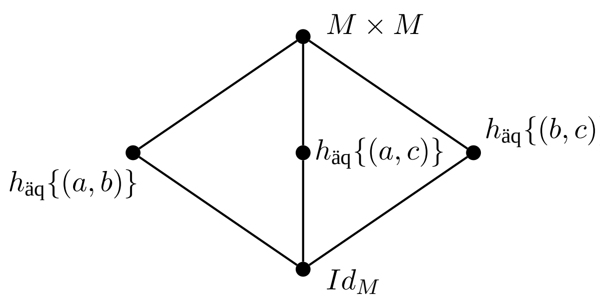
$$\lhd \;=\; < \setminus <^2, \text{ d.h. } x \lhd y \;\Leftrightarrow\; x < y \land \neg\exists z : x < z < y$$

gezeichnet. Für $x \lhd y$ sagt man auch, y ist ***oberer Nachbar*** von x. Weiterhin wird im Fall $x \lhd y$ der Knoten y oberhalb vom Knoten x gezeichnet, so dass auf Pfeilspitzen verzichtet werden kann.

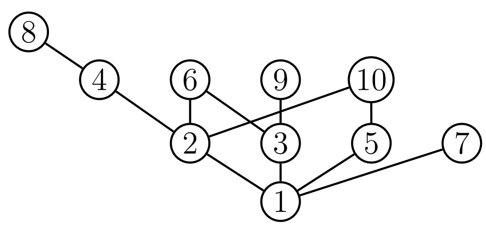
Beispiel 1.36

1. Die Inklusionsrelation auf der Potenzmenge $\mathcal{P}(M)$ von $M = \{a, b, c\}$ und die Verfeinerungsrelation auf der Menge $\mathcal{A}(M)$ aller Äquivalenzrelationen auf $M = \{a, b, c\}$ lassen sich durch folgende Hasse-Diagramme darstellen.





2. Die "teilt"-Relation auf $\{1, 2, \ldots, 10\}$ ist durch folgendes Hasse-Diagramme darstellbar.



Definition 1.37 *Sei $\leq$ eine Ordnung auf A und sei $h \in H$ Element einer Teilmenge $H \subseteq A$.*

- *h heißt **kleinstes Element** oder **Minimum** von H ($h = \min H$), falls gilt:*

$$\forall h' \in H : h \leq h'.$$

- *h heißt **größtes Element** oder **Maximum** von H ($h = \max H$), falls gilt:*

$$\forall h' \in H : h' \leq h.$$

- *a heißt **minimal** in H, falls es in H kein kleineres Element gibt:*

$$\forall h' \in H : h' \leq h \Rightarrow h' = h.$$

- *a heißt **maximal** in H, falls es in H kein größeres Element gibt:*

$$\forall h' \in H : h \leq h' \Rightarrow h = h'.$$

Bemerkung 1.38 *Wegen der Antisymmetrie kann es in H höchstens ein kleinstes und höchstens ein größtes Element geben.*

Definition 1.39 *Sei $\leq$ eine Ordnung auf A und sei $H \subseteq A$.*

- *Jedes Element $u \in A$ mit $u \leq h$ für alle $h \in H$ heißt **untere** und jedes $o \in A$ mit $h \leq o$ für alle $h \in H$ heißt **obere Schranke** von H.*
- *H heißt **nach oben beschränkt**, wenn H eine obere Schranke hat, und **nach unten beschränkt**, wenn H eine untere Schranke hat.*
- *H heißt **beschränkt**, wenn H nach oben und nach unten beschränkt ist.*
- *Besitzt H eine größte untere Schranke i, d.h. besitzt die Menge U aller unteren Schranken von H ein größtes Element i, so heißt i das **Infimum** von H ($i = \inf H$):*

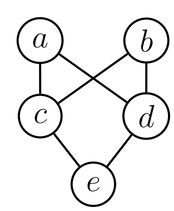
$$(\forall h \in H : h \geq i) \wedge [\forall u \in A : (\forall h \in H : h \geq u) \Rightarrow u \leq i].$$

- *Besitzt H eine kleinste obere Schranke s, d.h. besitzt die Menge O aller oberen Schranken von H ein kleinstes Element s, so heißt s das **Supremum** von H ($s = \sup H$):*

$$(\forall h \in H : h \leq s) \wedge [\forall o \in A : (\forall h \in H : h \leq o) \Rightarrow s \leq o]$$

Bemerkung 1.40 *H kann nicht mehr als ein Supremum und ein Infimum haben.*

Beispiel 1.41 Betrachte nebenstehende Ordnung auf der Menge $A = \{a, b, c, d, e\}$. Die folgende Tabelle zeigt für verschiedene Teilmengen $H \subseteq A$ alle minimalen und maximalen Elemente in H, alle unteren und oberen Schranken, sowie Minimum, Maximum, Infimum und Supremum von H (falls existent).



H	minimal	maximal	Minimum	Maximum	unt. Schr.	ob. Schr.	Inf.	Sup.
$\{a, b\}$	a, b	a, b	-	-	c, d, e	-	-	-
$\{c, d\}$	c, d	c, d	-	-	e	a, b	e	-
$\{a, b, c\}$	c	a, b	c	-	c, e	-	c	-
$\{a, b, c, e\}$	e	a, b	e	-	e	-	e	-
$\{a, c, d, e\}$	e	a	e	a	e	a	e	a

Bemerkung 1.42

- *Auch in linearen Ordnungen muss nicht jede beschränkte Teilmenge ein Supremum oder Infimum besitzen.*
- *So hat in der linear geordneten Menge* $(\mathbb{Q}, \leq)$ *die Teilmenge*

$$H = \{x \in \mathbb{Q} \mid x^2 \leq 2\}$$

weder ein Supremum noch ein Infimum.

- *Dagegen hat in einer linearen Ordnung jede* endliche *Teilmenge ein kleinstes und ein größtes Element und somit erst recht ein Supremum und ein Infimum.*

1.4.3 Abbildungen

Definition 1.43 *Sei* R *eine binäre Relation auf einer Menge* M. R *heißt* ***rechtseindeutig****, falls gilt:*

$$\forall x, y, z \in M : xRy \land xRz \Rightarrow y = z.$$

R *heißt* ***linkseindeutig****, falls gilt:*

$$\forall x, y, z \in M : xRz \land yRz \Rightarrow x = y.$$

Der ***Nachbereich*** $N(R)$ *und der* ***Vorbereich*** $V(R)$ *von* R *sind*

$$N(R) = \bigcup_{x \in M} R(x) \text{ und } V(R) = \bigcup_{x \in M} R^T(x).$$

Eine rechtseindeutige Relation R *mit* $V(R) = A$ *und* $N(R) \subseteq B$ *heißt* ***Abbildung*** *oder* ***Funktion von A nach B*** *(*kurz $R : A \to B$*).*

Bemerkung 1.44

- *Wie üblich werden wir Abbildungen meist mit kleinen Buchstaben* $f, g, h, \ldots$ *bezeichnen und für* $(x, y) \in f$ *nicht* xfy *sondern* $f(x) = y$ *oder* $f : x \mapsto y$ *schreiben.*
- *Ist* $f : A \to B$ *eine Abbildung, so wird der Vorbereich* $V(f) = A$ *der* ***Definitionsbereich*** *und die Menge* B *der* ***Wertebereich*** *oder* ***Wertevorrat*** *von* f *genannt.*
- *Der Nachbereich* $N(f)$ *wird als* ***Bild*** *von* f *bezeichnet.*

Definition 1.45

- *Im Fall* $N(f) = B$ *heißt* f ***surjektiv****.*

- *Ist f linkseindeutig, so heißt f **injektiv**. In diesem Fall impliziert $f(x) = f(y)$ die Gleichheit $x = y$.*
- *Eine injektive und surjektive Abbildung heißt **bijektiv**.*
- *Für eine injektive Abbildung $f : A \to B$ ist auch f^T eine Abbildung, die mit f^{-1} bezeichnet und die **inverse Abbildung** zu f genannt wird.*

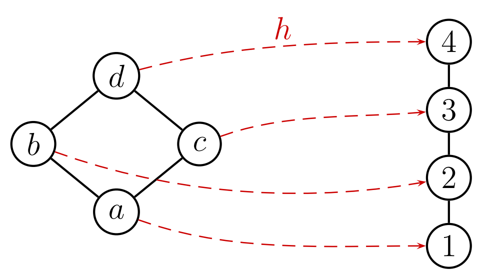
Man beachte, dass der Definitionsbereich $V(f^{-1}) = N(f)$ nur dann gleich B ist, wenn f auch surjektiv, also eine Bijektion ist.

1.4.4 Homo- und Isomorphismen

Definition 1.46 *Seien (A_1, R_1) und (A_2, R_2) Relationalstrukturen.*

- *Eine Abbildung $h : A_1 \to A_2$ heißt **Homomorphismus**, falls für alle $a, b \in A_1$ gilt:*
$$aR_1b \Rightarrow h(a)R_2h(b).$$
- *Sind (A_1, R_1) und (A_2, R_2) Ordnungen, so spricht man von **Ordnungshomomorphismen** oder einfach von **monotonen** Abbildungen.*
- *Injektive Ordnungshomomorphismen werden auch **streng monotone** Abbildungen genannt.*

Beispiel 1.47 Folgende Abbildung $h : A_1 \to A_2$ ist ein bijektiver Ordnungshomomorphismus.



$(A_1, \leq_1)$ $(A_2, \leq_2)$

Obwohl h ein bijektiver Homomorphismus ist, ist die Umkehrung h^{-1} kein Homomorphismus, da h^{-1} nicht monoton ist. Es gilt nämlich

$$2 \leq_2 3, \text{ aber } h^{-1}(2) = b \not\leq_1 c = h^{-1}(3).$$

Definition 1.48 *Ein bijektiver Homomorphismus $h : A_1 \to A_2$, bei dem auch h^{-1} ein Homomorphismus ist, d.h. es gilt*

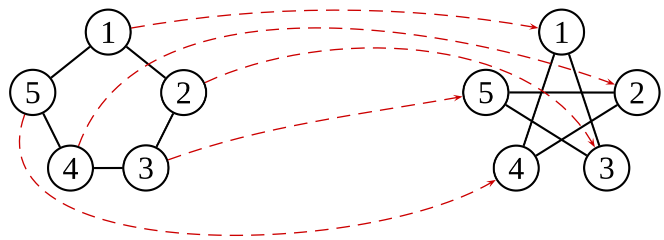
$$\forall a, b \in A_1 : aR_1b \Leftrightarrow h(a)R_2h(b).$$

*heißt **Isomorphismus**. In diesem Fall heißen die Strukturen (A_1, R_1) und (A_2, R_2) **isomorph** (kurz: $(A_1, R_1) \cong (A_2, R_2)$).*

Beispiel 1.49

- Die beiden folgenden Graphen sind isomorph. Zwei Isomorphismen sind beispielsweise h_1 und h_2.

v	1 2 3 4 5
$h_1(v)$	1 3 5 2 4
$h_2(v)$	1 4 2 5 3



- Die Bijektion $h : x \mapsto e^x$ ist ein Ordnungsisomorphismus zwischen $(\mathbb{R}, \leq)$ und $((0, \infty), \leq)$.

- Für $n \in \mathbb{N}$ sei

$$T_n \;=\; \{k \in \mathbb{N} \mid k \text{ teilt } n\}$$

  die Menge aller Teiler von n und $P_n \;=\; T_n \cap \text{Prim}$ die Menge aller Primteiler von n. Dann ist für quadratfreies n, d.h. es gibt kein $k \geq 2$, so dass k^2 die Zahl n teilt, die Abbildung

$$h : k \mapsto P_k$$

  ein Ordnungsisomorphismus zwischen $(T_n, |)$ und $(\mathcal{P}(P_n), \subseteq)$.

- Während auf der Knotenmenge $V = [3]$ insgesamt $2^3 = 8$ verschiedene Graphen existieren, gibt es auf dieser Menge nur 4 unterschiedliche nichtisomorphe Graphen:

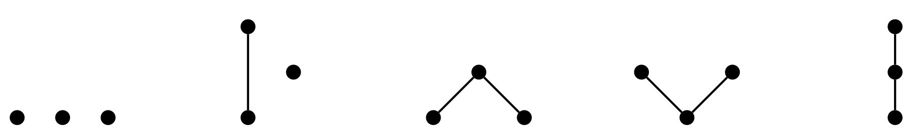
Bemerkung 1.50 *Auf der Knotenmenge $V = [n]$ existieren genau $2^{\binom{n}{2}}$ verschiedene Graphen. Sei $a(n)$ die Anzahl aller nichtisomorphen Graphen auf V. Da maximal $n!$ verschiedene Graphen auf V in einer Isomorphieklasse liegen können, ist $a(n) \geq 2^{\binom{n}{2}}/n!$.*

*Tatsächlich ist $a(n)$ **asymptotisch gleich** $g(n) = 2^{\binom{n}{2}}/n!$ (in Zeichen: $a(n) \sim g(n)$), d.h.*

$$\lim_{n\to\infty} a(n)/g(n) = 1.$$

Also gibt es auf $V = [n]$ nicht wesentlich mehr als $g(n)$ nichtisomorphe Graphen.

Beispiel 1.51 Es existieren genau 5 nichtisomorphe Ordnungen mit 3 Elementen:



Anders ausgedrückt: Die Klasse aller dreielementigen Ordnungen zerfällt unter der Äquivalenzrelation $\cong$ in fünf Äquivalenzklassen, die durch obige fünf Hasse-Diagramme repräsentiert werden.

1.5 Minimierung von DFAs

Wie können wir feststellen, ob ein DFA $M = (Z, \Sigma, \delta, q_0, E)$ unnötige Zustände enthält? Zunächst einmal können alle Zustände entfernt werden, die nicht vom Startzustand aus erreichbar sind. Im folgenden gehen wir daher davon aus, dass M keine unerreichbaren Zustände enthält. Offensichtlich können zwei Zustände q und p zu einem Zustand verschmolzen werden (kurz: $q \sim p$), wenn M von q und von p ausgehend jeweils dieselben Wörter akzeptiert. Bezeichnen wir den DFA $(Z, \Sigma, \delta, q, E)$ mit M_q und $L(M_q)$ mit L_q, so sind q und p genau dann verschmelzbar, wenn $L_q = L_p$ ist.
Fassen wir alle zu einem Zustand z äquivalenten Zustände in dem neuen Zustand

$$[z]_\sim = \{z' \in Z \mid L_{z'} = L_z\}$$

zusammen (wofür wir auch kurz $[z]$ oder $\tilde{z}$ schreiben) und ersetzen wir Z und E durch $\tilde{Z} = \{\tilde{z} \mid z \in Z\}$ und $\tilde{E} = \{\tilde{z} \mid z \in E\}$, so erhalten wir den DFA $M' = (\tilde{Z}, \Sigma, \delta', [q_0], \tilde{E})$ mit

$$\delta'([q], a) = [\delta(q, a)].$$

Im nächsten Satz zeigen wir, dass M' tatsächlich der gesuchte Minimalautomat für $L(M)$ ist. Für eine Teilmenge $Q \subseteq Z$ bezeichne $\tilde{Q}$ die Menge $\{\tilde{q} \mid q \in Q\}$ aller Äquivalenzklassen $\tilde{q}$, die einen Repräsentanten q in Q haben.

Theorem 1.52 *Sei $M = (Z, \Sigma, \delta, q_0, E)$ ein DFA, der nur Zustände enthält, die vom Startzustand q_0 aus erreichbar sind. Dann ist $M' = (\tilde{Z}, \Sigma, \delta', [q_0], \tilde{E})$ mit*

$$\delta'([q], a) = [\delta(q, a)]$$

ein DFA für $L(M)$ mit einer minimalen Anzahl von Zuständen.

Beweis

- Wir zeigen zuerst, dass δ' wohldefiniert ist, also der Wert von $\delta'(\tilde{q}, a)$ nicht von der Wahl des Repräsentanten q abhängt. Hierzu zeigen wir, dass im Fall $p \sim q$ auch $\delta(q, a)$ und $\delta(p, a)$ äquivalent sind:

$$\begin{aligned} L_q = L_p \quad &\Rightarrow \quad \forall x \in \Sigma^* : x \in L_q \leftrightarrow x \in L_p \\ &\Rightarrow \quad \forall x \in \Sigma^* : ax \in L_q \leftrightarrow ax \in L_p \\ &\Rightarrow \quad \forall x \in \Sigma^* : x \in L_{\delta(q,a)} \leftrightarrow x \in L_{\delta(p,a)} \\ &\Rightarrow \quad L_{\delta(q,a)} = L_{\delta(p,a)}. \end{aligned}$$

- Als nächstes zeigen wir, dass $L(M') = L(M)$ ist. Sei $x = x_1 \cdots x_n$ eine Eingabe und seien
$$q_i = \hat{\delta}(q_0, x_1 \cdots x_i),\ i = 0, \ldots, n$$
die von M beim Abarbeiten von x durchlaufenen Zustände. Wegen
$$\delta'([q_{i-1}], x_i) = [\delta(q_{i-1}, x_i)] = [q_i]$$
durchläuft M' dann die Zustände
$$[q_0], [q_1], \cdots, [q_n].$$
Da aber q_n genau dann zu E gehört, wenn $[q_n] \in \tilde{E}$ ist, folgt $L(M') = L(M)$.

- Es bleibt zu zeigen, dass M' eine minimale Anzahl $\|\tilde{Z}\|$ von Zuständen hat. Dies ist sicher dann der Fall, wenn bereits M minimal ist. Es reicht also zu zeigen, dass die Anzahl $k = \|\tilde{Z}\| = \|\{L_q \mid q \in Z\}\|$ nicht von M, sondern nur von $L(M)$ abhängt. Für $x \in \Sigma^*$ sei
$$L_x = \{y \in \Sigma^* \mid xy \in L(M)\}.$$
Dann gilt $\{L_x \mid x \in \Sigma^*\} \subseteq \{L_q \mid q \in Z\}$, da $L_x = L_{\hat{\delta}(q_0,x)}$ ist. Die umgekehrte Inklusion gilt ebenfalls, da nach Voraussetzung jeder Zustand $q \in Z$ über ein $x \in \Sigma^*$ erreichbar ist. Also hängt $k = \|\{L_q \mid q \in Z\}\| = \|\{L_x \mid x \in \Sigma^*\}\|$ nur von $L(M)$ ab.

■