Vorlesungsskript

Einführung in die Theoretische Informatik

Wintersemester 2016/17

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

30. November 2017

Inhaltsverzeichnis

Inhaltsverzeichnis

1 Einleitung

Rechenmaschinen spielen in der Informatik eine zentrale Rolle. In dieser Vorlesung beschäftigen wir uns mit mathematischen Modellen für Maschinentypen von unterschiedlicher Berechnungskraft. Unter anderem lernen wir das Rechenmodell der Turingmaschine (TM) kennen, mit dem sich alle anderen Rechenmodelle simulieren lassen. Ein weiteres wichtiges Thema der Vorlesung ist die Frage, welche Probleme algorithmisch lösbar sind und wo die Grenzen der Berechenbarkeit verlaufen.

Schließlich untersuchen wir die Komplexität von algorithmischen Problemen, indem wir den benötigten Rechenaufwand möglichst gut nach oben und unten abschätzen. Eine besondere Rolle spielen hierbei die NP-vollständigen Probleme, deren Komplexität bis heute offen ist.

Themen der Vorlesung

- Welche Rechenmodelle sind für bestimmte Aufgaben adäquat? (Automatentheorie)
- Welche Probleme sind lösbar? (Berechenbarkeitstheorie)
- Welcher Aufwand ist zur Lösung eines algorithmischen Problems nötig? (Komplexitätstheorie)

In den theoretisch orientierten Folgeveranstaltungen wird es dagegen um folgende Themen gehen.

Thema der Vorlesung Algorithmen und Datenstrukturen

- Wie lassen sich praktisch relevante Problemstellungen möglichst effizient lösen? (Algorithmik)

Thema der Vorlesung Logik in der Informatik

- Mathematische Grundlagen der Informatik, Beweise führen, Modellierung (Aussagenlogik, Prädikatenlogik)

Die wichtigsten **Lernziele der Vorlesung** sind:

- Überblick über die wichtigsten Rechenmodelle (Automaten) wie z.B.
  - endliche Automaten
  - Kellerautomaten
  - Turingmaschinen
  - Registermaschinen
  - Schaltkreise
- Charakterisierung der Klassen aller mit diesen Rechenmodellen lösbaren Probleme durch
  - unterschiedliche Typen von formalen Grammatiken
  - Abschlusseigenschaften unter geeigneten Sprachoperationen
  - Reduzierbarkeit auf typische Probleme (Vollständigkeit)
- Erkennen von Grenzen der Berechenbarkeit
- Klassifikation wichtiger algorithmischer Probleme nach ihrer Komplexität

Rechenmaschinen spielen in der Informatik eine zentrale Rolle Es gibt viele unterschiedliche mathematische Modelle für Rechenmaschinen. Diese können sich in ihrer Berechnungskraft unterscheiden Die Turingmaschine (TM) ist ein universales Berechnungsmodell, da sie alle anderen bekannten Rechenmodelle simulieren kann. Wir betrachten zunächst Einschränkungen des TM-Modells, die vielfältige praktische Anwendungen haben, wie z.B.

- endliche Automaten (DFA, NFA)
- Kellerautomaten (PDA, DPDA) etc.

Der Begriff *Algorithmus* geht auf den persischen Gelehrten Muhammed

Al Chwarizmi (8./9. Jhd.) zurück. Der älteste bekannte nicht-triviale Algorithmus ist der nach *Euklid* benannte Algorithmus zur Berechnung des größten gemeinsamen Teilers zweier natürlicher Zahlen (300 v. Chr.). Von einem Algorithmus wird erwartet, dass er für jede zulässige *Problemeingabe* nach endlich vielen Rechenschritten eine korrekte *Ausgabe* liefert. Eine wichtige Rolle spielen *Entscheidungsprobleme*, bei denen jede Eingabe nur mit ja oder nein beantwortet wird. Die (maximale) Anzahl der Rechenschritte bei allen möglichen Eingaben ist nicht beschränkt, d.h. mit wachsender Eingabelänge kann auch die Rechenzeit beliebig anwachsen. Die Beschreibung eines Algorithmus muss jedoch endlich sein. Problemeingaben können Zahlen, Formeln, Graphen etc. sein. Diese werden über einem *Eingabealphabet* Σ kodiert.

Definition 1.

a) Ein ***Alphabet*** $\Sigma = \{a_1, \dots, a_m\}$ *ist eine geordnete Menge von endlich vielen* ***Zeichen****.*

b) Eine Folge $x = x_1 \dots x_n$ *von* n *Zeichen heißt* ***Wort*** *(der* ***Länge*** $|x| = n$*).*

c) Die Menge aller Wörter über Σ *ist*

$$\Sigma^* = \bigcup_{n \geq 0} \Sigma^n,$$

wobei $\Sigma^n = \{x_1 \dots x_n \mid n \geq 0$ *und* $x_i \in \Sigma$ *für* $i = 1, \dots, n\}$ *alle Wörter der Länge* n *enthält.*

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

e) Jede Teilmenge $L \subseteq \Sigma^*$ *heißt* ***Sprache*** *über dem Alphabet* Σ*.*

Beispiel 2. *Sei* Σ *ein Alphabet. Dann sind* $\varnothing, \Sigma^*, \Sigma$ *und* $\{\varepsilon\}$ *Sprachen über* Σ*. Die Sprache* $\varnothing$ *enthält keine Wörter und heißt* ***leere Sprache****. Die Sprache* Σ^* *enthält dagegen alle Wörter über* Σ*, während die Sprache* Σ *alle Wörter über* Σ *der Länge 1 enthält. Die Sprache* $\{\varepsilon\}$ *enthält nur das leere Wort, ist also einelementig. Einelementige Sprachen werden auch als* ***Singletonsprachen*** *bezeichnet.*

Da Sprachen Mengen sind, können wir sie bzgl. Inklusion vergleichen. Zum Beispiel gilt

$$\varnothing \subseteq \{\varepsilon\} \subseteq \Sigma^*.$$

Wir können Sprachen auch vereinigen, schneiden und komplementieren. Seien A und B Sprachen über Σ. Dann ist

- $A \cap B = \{x \in \Sigma^* \mid x \in A, x \in B\}$ der ***Schnitt*** von A und B,
- $A \cup B = \{x \in \Sigma^* \mid x \in A \vee x \in B\}$ die ***Vereinigung*** von A und B, und
- $\overline{A} = \{x \in \Sigma^* \mid x \notin A\}$ das ***Komplement*** von A.

Neben den Mengenoperationen gibt es auch spezielle Sprachoperationen.

Definition 3.

- *Das* ***Produkt*** *(****Verkettung****,* ***Konkatenation****) der Sprachen* A *und* B *ist*

$$AB = \{xy \mid x \in A, y \in B\}.$$

Ist $A = \{x\}$ *eine Singletonsprache, so schreiben wir für* $\{x\}B$ *auch einfach* xB*.*

- *Die* ***n-fache Potenz*** $\boldsymbol{A^n}$ *einer Sprache* A *ist induktiv definiert durch*

$$A^n = \begin{cases} \{\varepsilon\}, & n = 0, \\ A^{n-1}A, & n > 0. \end{cases}$$

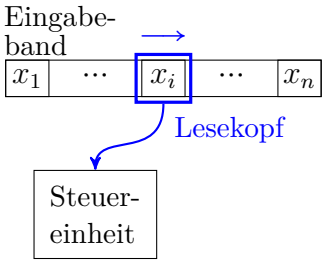
- *Die* ***Sternhülle*** $\boldsymbol{A^*}$ *von* A *ist* $A^* = \bigcup_{n \geq 0} A^n$ *und die* ***Plushülle*** $\boldsymbol{A^+}$ *von* A *ist* $A^+ = \bigcup_{n \geq 1} A^n = AA^*$*.*

2 Reguläre Sprachen

Wir betrachten zunächst Einschränkungen des TM-Modells, die vielfältige praktische Anwendungen haben, wie z.B. endliche Automaten (DFA, NFA), Kellerautomaten (PDA, DPDA) etc.

2.1 Endliche Automaten

Ein endlicher Automat führt bei einer Eingabe der Länge n nur n Rechenschritte aus. Um die gesamte Eingabe lesen zu können, muss der Automat also in jedem Schritt ein Zeichen der Eingabe verarbeiten.



Definition 4. *Ein* ***endlicher Automat*** *(kurz: DFA; deterministic finite automaton) wird durch ein 5-Tupel $M = (Z, \Sigma, \delta, q_0, E)$ beschrieben, wobei*

- *$Z \neq \varnothing$ 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*

$$L(M) = \left\{ x_1 \dots x_n \in \Sigma^* \;\middle|\; \begin{array}{l} \text{es gibt } q_1, \dots, q_{n-1} \in Z, q_n \in E \text{ mit} \\ \delta(q_i, x_{i+1}) = q_{i+1} \text{ für } i = 0, \dots, n-1 \end{array} \right\}.$$

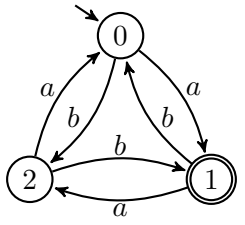
Eine Zustandsfolge $q_0, q_1, \dots, q_n$ heißt ***Rechnung*** *von $M(x_1 \dots x_n)$, falls $\delta(q_i, x_{i+1}) = q_{i+1}$ für $i = 0, \dots, n-1$ gilt. Sie heißt* ***akzeptierend****, falls $q_n \in E$ ist, und andernfalls* ***verwerfend****. Eine von einem DFA akzeptierte Sprache wird als* ***regulär*** *bezeichnet. Die zugehörige Sprachklasse ist*

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

Beispiel 5. *Betrachte den DFA $M = (Z, \Sigma, \delta, 0, E)$ mit $Z = \{0, 1, 2\}$, $\Sigma = \{a, b\}$, $E = \{1\}$ und der Überführungsfunktion*

δ	0	1	2
a	1	2	0
b	2	0	1

Graphische Darstellung:



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

Bei Eingabe $w_1 = aba$ führt M die akzeptierende Rechnung $0, 1, 0, 1$ durch, d.h. $w_1 \in L(M)$. Dagegen verwirft M das Wort $w_2 = abba$ (verwerfende Rechnung: $0, 1, 0, 2, 0$). ◁

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

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

Die von M erkannte Sprache lässt sich nun elegant durch

$$L(M) = \{x \in \Sigma^* \mid \hat{\delta}(q_0, x) \in E\}$$

beschreiben.

Behauptung 6. *Der DFA M aus Beispiel 5 akzeptiert die Sprache*

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

wobei $\#_a(x)$ die Anzahl der Vorkommen des Zeichens a in x bezeichnet und $i \equiv_m j$ (in Worten: i ist kongruent zu j modulo m) bedeutet, dass $i - j$ durch m teilbar ist.

Beweis. Da M nur den Endzustand 1 hat, ist $L(M) = \{x \in \Sigma^* \mid \hat{\delta}(0, x) = 1\}$, d.h. wir müssen folgende Äquivalenz zeigen:

$$\hat{\delta}(0, x) = 1 \Leftrightarrow \#_a(x) - \#_b(x) \equiv_3 1.$$

Hierzu reicht es, die Kongruenz

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

zu beweisen, wofür wir Induktion über die Länge n von x benutzen.

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

Induktionsschritt ***($n \rightsquigarrow n+1$):*** Sei $x = x_1 \dots x_{n+1}$ gegeben und sei $i = \hat{\delta}(0, x_1 \dots x_n)$. Nach IV gilt dann

$$i \equiv_3 \#_a(x_1 \dots x_n) - \#_b(x_1 \dots x_n).$$

Wegen $\delta(i, a) \equiv_3 i + 1$ und $\delta(i, b) \equiv_3 i - 1$ folgt daher

$$\begin{aligned} \delta(i, x_{n+1}) &\equiv_3 i + \#_a(x_{n+1}) - \#_b(x_{n+1}) \\ &\equiv_3 \#_a(x_1 \dots x_n) - \#_b(x_1 \dots x_n) + \#_a(x_{n+1}) - \#_b(x_{n+1}) \\ &= \#_a(x) - \#_b(x). \end{aligned}$$

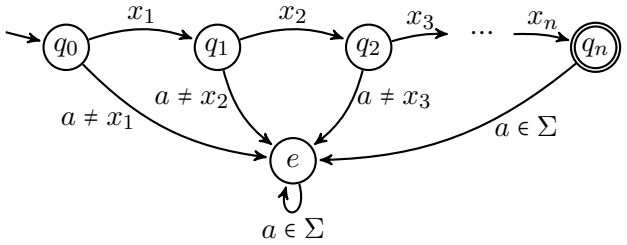
und somit

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

■

Beobachtung 7. *Alle Singletonsprachen sind regulär.*

Beweis. Für jedes Wort $x = x_1 \dots x_n$ existiert ein DFA M_x mit $L(M_x) = \{x\}$:



Formal ist M_x also das Tupel $(Z, \Sigma, \delta, q_0, E)$ mit $Z = \{q_0, \dots, q_n, e\}$, $E = \{q_n\}$ und der Überführungsfunktion

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

■

Als nächstes betrachten wir Abschlusseigenschaften der Sprachklasse REG.

Definition 8. *Ein* ***k-stelliger Sprachoperator*** *ist eine Abbildung op, die k Sprachen $L_1, \dots, L_k$ auf eine Sprache $op(L_1, \dots, L_k)$ abbildet.*

Beispiel 9. *Der Schnittoperator $\cap$ bildet zwei Sprachen L_1 und L_2 auf die Sprache $L_1 \cap L_2$ ab.* ◁

Definition 10. *Eine Sprachklasse $\mathcal{K}$ heißt unter op* ***abgeschlossen****, wenn gilt:*

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

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

Beispiel 11. *Der Abschluss der Singletonsprachen unter $\cap$ besteht aus allen Singletonsprachen und der leeren Sprache.*

Der Abschluss der Singletonsprachen unter $\cup$ besteht aus allen nichtleeren endlichen Sprachen.

Der Abschluss der Singletonsprachen unter $\cap$, $\cup$ und Komplement besteht aus allen endlichen und co-endlichen Sprachen.[*] ◁

Definition 12. *Für eine Sprachklasse $\mathcal{C}$ bezeichne co-$\mathcal{C}$ die Klasse $\{\bar{L} \mid L \in \mathcal{C}\}$ aller Komplemente von Sprachen in $\mathcal{C}$.*

Es ist leicht zu sehen, dass $\mathcal{C}$ genau dann unter Komplementbildung abgeschlossen ist, wenn *co-*$\mathcal{C} = \mathcal{C}$ ist.

Beobachtung 13. *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.*

Beweis. Sind $M_i = (Z_i, \Sigma, \delta_i, q_0, E_i)$, $i = 1, 2$, 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?) ■

[*] Eine Sprache $L \subseteq \Sigma^*$ ist co-endlich, wenn ihr Komplement $\bar{L}$ endlich ist.

Aus Beobachtung 13 folgt, dass alle endlichen und alle co-endlichen Sprachen regulär sind. Da die in Beispiel 5 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 Produkt oder Sternhülle abgeschlossen ist. Im übernächsten Abschnitt werden wir sehen, dass die Klasse REG als der Abschluss der endlichen Sprachen unter Vereinigung, Produkt und Sternhülle charakterisierbar (und somit auch unter diesen Operationen abgeschlossen) 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 lösen, da dieser den richtigen Zeitpunkt „erraten“ kann.

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

2.2 Nichtdeterministische endliche Automaten

Definition 14. *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 die Form*

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

hat. Hierbei bezeichnet $\mathcal{P}(Z)$ die **Potenzmenge** *(also die Menge aller Teilmengen) von Z. Diese wird auch oft mit 2^Z bezeichnet. Die von N akzeptierte Sprache ist*

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

Eine Zustandsfolge $q_0, q_1, \dots, q_n$ heißt **Rechnung** *von $N(x_1 \dots x_n)$, falls $q_{i+1} \in \Delta(q_i, x_{i+1})$ für $i = 0, \dots, n-1$ gilt.*

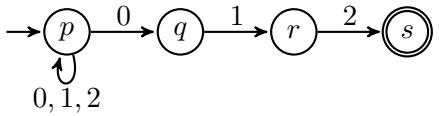
Ein NFA N kann bei einer Eingabe x also nicht nur eine, sondern mehrere verschiedene Rechnungen parallel ausführen. Ein Wort x gehört genau dann zu $L(N)$, wenn $N(x)$ mindestens eine akzeptierende Rechnung hat.

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 $\Delta(q, x_i) = \varnothing$ nicht gelesen werden kann.

Beispiel 15. *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	$\{p, q\}$	$\varnothing$	$\varnothing$	$\varnothing$
1	$\{p\}$	$\{r\}$	$\varnothing$	$\varnothing$
2	$\{p\}$	$\varnothing$	$\{s\}$	$\varnothing$

Graphische Darstellung:



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

Beobachtung 16. *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.*

Beweis. Sei $L_i = L(N_i)$. Wir können $Z_1 \cap Z_2 = \varnothing$ annehmen. Dann akzeptiert der NFA

$$N = (Z_1 \cup Z_2, \Sigma, \Delta_3, Q_1, E)$$

mit

$$\Delta_3(p, a) = \begin{cases} \Delta_1(p, a), & p \in Z_1 \setminus E_1, \\ \Delta_1(p, a) \cup \bigcup_{q \in Q_2} \Delta_2(q, a), & p \in E_1, \\ \Delta_2(p, a), & \text{sonst} \end{cases}$$

und

$$E = \begin{cases} E_2, & Q_2 \cap E_2 = \varnothing \\ E_1 \cup E_2, & \text{sonst} \end{cases}$$

die Sprache $L_1 L_2$.

$L_1 L_2 \subseteq L(N)$: Seien $x = x_1 \cdots x_k \in L_1$, $y = y_1 \cdots y_l \in L_2$ und seien $q_0, \dots, q_k$ und $p_0, \dots, p_l$ akzeptierende Rechnungen von $N_1(x)$ und $N_2(y)$. Dann ist $q_0, \dots, q_k, p_1, \dots, p_l$ eine akz. Rechnung von $N(xy)$, da $q_0 \in Q_1$ und $p_l \in E_2$ ist, und

- im Fall $l \geq 1$ wegen $q_k \in E_1$, $p_0 \in Q_2$ und $p_1 \in \Delta_2(p_0, y_1)$ zudem $p_1 \in \Delta(q_k, y_1)$ und
- im Fall $l = 0$ wegen $q_k \in E_1$ und $p_l \in Q_2 \cap E_2$ zudem $q_k \in E$ ist.

$L(N) \subseteq L_1 L_2$: Sei $x = x_1 \cdots x_n \in L(N)$ und sei $q_0, \dots, q_n$ eine akz. Rechnung von $N(x)$. Dann gilt $q_0 \in Q_1$, $q_n \in E$, $q_0, \dots, q_i \in Z_1$ und $q_{i+1}, \dots, q_n \in Z_2$ für ein $i \leq n$. Wir zeigen, dass ein $q \in Q_2$ existiert, so dass $q_0, \dots, q_i$ eine akz. Rechnung von $N_1(x_1 \cdots x_i)$ und $q, q_{i+1}, \dots, q_n$ eine akz. Rechnung von $N_2(x_{i+1} \cdots x_n)$ ist.

- Im Fall $i < n$ impliziert der Übergang $q_{i+1} \in \Delta(q_i, x_{i+1})$, dass $q_i \in E_1$ und $q_{i+1} \in \Delta_2(q, x_{i+1})$ für ein $q \in Q_2$ ist.
- Im Fall $i = n$ ist $q_n \in E_1$ und $Q_2 \cap E_2 \neq \varnothing$ (d.h. $\varepsilon \in L_2$).

Ganz ähnlich lässt sich zeigen, dass der NFA

$$N^* = (Z_1 \cup \{q_{neu}\}, \Sigma, \Delta_4, Q_1 \cup \{q_{neu}\}, E_1 \cup \{q_{neu}\})$$

mit

$$\Delta_4(p,a) = \begin{cases} \Delta_1(p,a), & p \in Z_1 \setminus E_1, \\ \Delta_1(p,a) \cup \bigcup_{q \in Q_1} \Delta_1(q,a), & p \in E_1, \\ \varnothing, & \text{sonst} \end{cases}$$

die Sprache L_1^* akzeptiert. ∎

Satz 17 (Rabin und Scott)**.**
$\mathsf{REG} = \{L(N) \mid N \textit{ ist ein NFA}\}$.

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 $M = (\mathcal{P}(Z), \Sigma, \delta, Q_0, E')$ mit $L(M) = L(N)$. Wir definieren die Überführungsfunktion $\delta : \mathcal{P}(Z) \times \Sigma \to \mathcal{P}(Z)$ von M mittels

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

Die Menge $\delta(Q,a)$ enthält also alle Zustände, in die N gelangen kann, wenn N ausgehend von einem beliebigen Zustand $q \in Q$ das Zeichen a liest. Intuitiv bedeutet dies, dass der DFA M den NFA N simuliert, indem M in seinem aktuellen Zustand Q die Information speichert, in welchen Zuständen sich N momentan befinden könnte. Für die Erweiterung $\hat{\delta} : \mathcal{P}(Z) \times \Sigma^* \to \mathcal{P}(Z)$ von δ (siehe Seite 4) können wir nun folgende Behauptung zeigen.

Behauptung. $\hat{\delta}(Q_0, x)$ enthält alle Zustände, die N ausgehend von einem Startzustand nach Lesen von x erreichen kann.

Wir beweisen die Behauptung induktiv über die Länge n von x.

Induktionsanfang ***($n = 0$):*** klar, da $\hat{\delta}(Q_0, \varepsilon) = Q_0$ ist.

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

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

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

Deklarieren wir nun diejenigen Teilmengen $Q \subseteq Z$, die mindestens einen Endzustand von N enthalten, als Endzustände des **Potenzmengenautomaten** M, d.h.

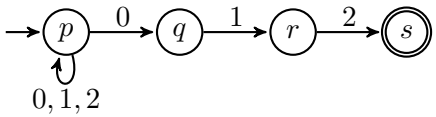
$$E' = \{Q \subseteq Z \mid Q \cap E \neq \varnothing\},$$

so folgt für alle Wörter $x \in \Sigma^*$:

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

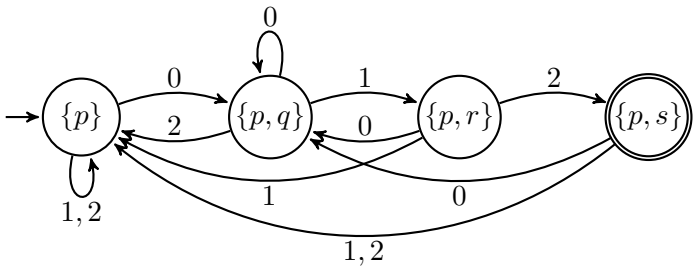
∎

Beispiel 18. *Für den NFA $N = (Z, \Sigma, \Delta, Q_0, E)$ aus Beispiel 15*



ergibt die Konstruktion des vorigen Satzes den folgenden DFA M (nach Entfernen 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 Potenzmengenautomaten benötigt werden (siehe Übungen).

Korollar 19. *Die Klasse* REG *der regulären Sprachen ist unter folgenden Operationen abgeschlossen:*

- *Komplement,*
- *Schnitt,*
- *Vereinigung,*
- *Produkt,*
- *Sternhülle.*

2.3 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 kennenlernen:

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

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

Definition 20. *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* $\varnothing$*,* ϵ *und* a *(*$a \in \Sigma$*) sind reguläre Ausdrücke, die*

- *die leere Sprache* $L(\varnothing) = \varnothing$*,*
- *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.

Bemerkung 21.

- *Um Klammern zu sparen, definieren wir folgende* ***Präzedenzordnung****: Der Sternoperator* * *bindet stärker als der Produktoperator und dieser wiederum stärker als der Vereinigungsoperator. Für* $((a|b(c)^*)|d)$ *können wir also kurz* $a|bc^*|d$ *schreiben.*
- *Da der reguläre Ausdruck* $\gamma\gamma^*$ *die Sprache* $L(\gamma)^+$ *beschreibt, verwenden wir* γ^+ *als Abkürzung für den Ausdruck* $\gamma\gamma^*$*.*

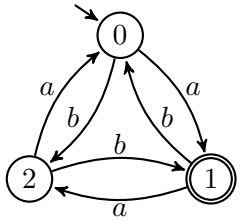
Beispiel 22. *Die regulären Ausdrücke* ϵ^*, $\varnothing^*$, $(0|1)^*00$ *und* $\epsilon 0|\varnothing 1^*$ *beschreiben folgende Sprachen:*

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

◁

Beispiel 23. *Betrachte nebenstehenden 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 $L_{0,0}$ *aller Wörter* x*, die den DFA* M *ausgehend vom Zustand* 0 *in den Zustand* 0 *überführen. Weiter sei* $L_{0,0}^{\neq 0}$ *die Sprache aller solchen Wörter* $w \in L_{0,0}$*, die den Zustand 0 nur zu Beginn und am Ende (aber nicht zwischendurch) besuchen. Dann setzt sich jedes* $x \in L_{0,0}$ *aus beliebig vielen Teilwörtern* $w_1, \ldots, w_k \in L_{0,0}^{\neq 0}$ *zusammen, d.h.* $L_{0,0} = (L_{0,0}^{\neq 0})^*$*.*

Jedes $w \neq \varepsilon$ *in* $L_{0,0}^{\neq 0}$ *beginnt entweder mit einem* a *(Übergang von* 0 *nach* 1*) oder mit einem* b *(Übergang von* 0 *nach* 2*). Im ersten Fall folgt eine beliebige Anzahl von Teilwörtern* ab *(Wechsel zwischen* 1 *und* 2*), an die sich entweder das Suffix* aa *(Rückkehr von* 1 *nach* 0 *über* 2*) oder das Suffix* b *(direkte Rückkehr von* 1 *nach* 0*) anschließt. Analog folgt im zweiten Fall eine beliebige Anzahl von Teilwörtern* ba *(Wechsel zwischen* 2 *und* 1*), an die sich entweder das Suffix* a *(direkte Rückkehr von* 2 *nach* 0*) oder das Suffix* bb *(Rückkehr von* 2 *nach* 0 *über* 1*) anschließt. Daher lässt sich* $L_{0,0}^{\neq 0}$ *durch den regulären Ausdruck*

$$\gamma_{0,0}^{\neq 0} = a(ab)^*(aa|b) \mid b(ba)^*(a|bb) \mid \epsilon$$

beschreiben. Eine ähnliche Überlegung zeigt, dass sich die die Sprache $L_{0,1}^{\neq 0}$ aller Wörter, die M ausgehend von 0 in den Zustand 1 überführen, ohne dass zwischendurch der Zustand 0 nochmals besucht wird, durch den regulären Ausdruck $\gamma_{0,1}^{\neq 0} = (a|bb)(ab)^*$ beschreibbar ist. Somit erhalten wir für $L(M)$ den regulären Ausdruck

$$\gamma_{0,1} = (a(ab)^*(aa|b) \mid b(ba)^*(a|bb))^*(a|bb)(ab)^*.$$

◁

Satz 24. $\{L(\gamma) \mid \gamma$ *ist ein regulärer Ausdruck*$\} = \mathsf{REG}$.

Beweis. Die Inklusion von rechts nach links ist klar, da die Basisausdrücke $\varnothing$, ϵ und a, $a \in \Sigma^*$, nur reguläre Sprachen beschreiben und die Sprachklasse REG unter Produkt, Vereinigung und Sternhülle abgeschlossen ist (siehe Beobachtungen 13 und 16).

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

$$L_{p,q}^r = \left\{ x_1 \ldots x_n \in \Sigma^* \;\middle|\; \begin{array}{c} \hat{\delta}(p, x_1 \ldots x_n) = q \text{ und für} \\ i = 1, \ldots, n-1 \text{ gilt } \hat{\delta}(p, x_1 \ldots x_i) \leq r \end{array} \right\}.$$

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

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

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

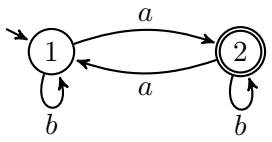
$$L^{r+1}_{p,q} \;=\; L^r_{p,q} \cup L^r_{p,r+1}(L^r_{r+1,r+1})^* L^r_{r+1,q}$$

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 25. *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 $\gamma^2_{1,2}$ zu berechnen, benutzen wir die Rekursionsformel

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

und erhalten

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

Um den regulären Ausdruck $\gamma^2_{1,2}$ für $L(M)$ zu erhalten, genügt es also, die regulären Ausdrücke $\gamma^0_{1,1}$, $\gamma^0_{1,2}$, $\gamma^0_{2,1}$, $\gamma^0_{2,2}$, $\gamma^1_{1,2}$ und $\gamma^1_{2,2}$ zu berechnen:

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

◁

Korollar 26. *Sei L eine Sprache. Dann sind folgende Aussagen äquivalent:*

- *L ist regulär (d.h. es gibt einen DFA M mit $L = L(M)$),*
- *es gibt einen NFA N mit $L = L(N)$,*
- *es gibt einen regulären Ausdruck γ mit $L = L(\gamma)$,*
- *L lässt sich mit den Operationen Vereinigung, Produkt und Sternhülle aus endlichen Sprachen gewinnen,*
- *L lässt sich mit den Operationen $\cap$, $\cup$, Komplement, Produkt und Sternhülle aus endlichen Sprachen gewinnen.*

Wir werden bald noch eine weitere Charakterisierung von REG kennenlernen, nämlich durch reguläre Grammatiken. Zuvor befassen wir uns jedoch mit dem Problem, DFAs zu minimieren. Dabei spielen Relationen (insbesondere Äquivalenzrelationen) eine wichtige Rolle.

2.4 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.

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 27.

- (F, M) *mit* $F = \{f \mid f$ *ist Fluss in Europa*$\}$ *und*
$$M = \{(f, g) \in F \times F \mid f \text{ mündet in } g\}.$$
- (U, B) *mit* $U = \{x \mid x$ *ist Berliner*$\}$ *und*
$$B = \{(x, y) \in U \times U \mid x \text{ ist Bruder von } y\}.$$
- $(\mathsf{P}(M), \subseteq)$*, 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 (d.h.* $a|b$*, falls ein* $c \in \mathbb{Z}$ *mit* $b = ac$ *existiert).* ◁

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

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

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:

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

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

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

R^T wird oft auch mit R^{-1} bezeichnet. Z.B. ist $(\mathbb{R}, \leq^T) = (\mathbb{R}, \geq)$.

Seien R und S Relationen auf A. Das ***Produkt*** oder die ***Komposition*** von R und S ist

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

Beispiel 28. *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 Onkel-Relation.* ◁

Übliche Bezeichnungen für das Relationenprodukt sind auch $R;S$ und $R \cdot S$ oder einfach RS. Das n-fache Relationenprodukt $R \circ \cdots \circ R$ von R wird mit R^n bezeichnet. Dabei ist $R^0 = Id$.

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

Eigenschaften von Relationen

Sei R eine Relation auf A. Dann heißt R

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

gilt.

Die nachfolgende Tabelle gibt einen Überblick über die wichtigsten Relationalstrukturen.

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

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

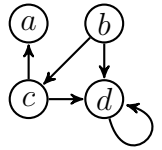
Beispiel 29.

- *Die Relation "ist Schwester von" ist zwar in einer reinen Damengesellschaft symmetrisch, i.a. jedoch weder symmetrisch noch asymmetrisch noch antisymmetrisch.*
- *Die Relation "ist Geschwister von" ist zwar symmetrisch, aber weder reflexiv noch transitiv und somit keine Äquivalenzrelation.*
- $(\mathbb{R}, <)$ *ist irreflexiv, asymmetrisch, transitiv und semikonnex und somit eine lineare Striktordnung.*
- $(\mathbb{R}, \leq)$ *und* $(\mathsf{P}(M), \subseteq)$ *sind reflexiv, antisymmetrisch und transitiv und somit Ordnungen.*
- $(\mathbb{R}, \leq)$ *ist auch konnex und somit eine lineare Ordnung.*
- $(\mathsf{P}(M), \subseteq)$ *ist zwar im Fall* $\|M\| \leq 1$ *konnex, aber im Fall* $\|M\| \geq 2$ *weder semikonnex noch konnex.* ◁

Graphische Darstellung von Relationen

Eine Relation R auf einer endlichen Menge A kann durch einen ***gerichteten Graphen*** (oder ***Digraphen***) $G = (V, E)$ 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 30. *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 $\deg^+(x) = \|R[x]\|$, wobei $R[x] = \{y \in V \mid xRy\}$ die Menge der ***Nachfolger*** von x ist. Entsprechend ist $\deg^-(x) = \|\{y \in V \mid yRx\}\|$ der ***Eingangsgrad*** von x und $R^{-1}[x] = \{y \in V \mid yRx\}$ die Menge der ***Vorgänger*** von x. Falls R symmetrisch ist, werden die Pfeilspitzen meist weggelassen. In diesem Fall ist $d(x) = \deg^-(x) = \deg^+(x)$ der ***Grad*** von x und $R[x] = R^{-1}[x]$ heißt die ***Nachbarschaft*** von x. Ist R zudem irreflexiv, so ist G ***schleifenfrei*** und wir erhalten einen ***(ungerichteten) Graphen***. Eine irreflexive und symmetrische Relation R wird meist als Menge der ungeordneten Paare $E = \big\{\{a,b\} \mid aRb\big\}$ notiert.

Darstellung durch Adjazenzmatrizen

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 Adjazenzlisten

Eine weitere Möglichkeit besteht darin, eine endliche Relation R in Form einer Tabelle darzustellen, die jedem Element $x \in A$ seine Nachfolger in Form einer Liste zuordnet. Für obige Relation R erhalten wir folgende Listen:

x:	$R[x]$
a:	-
b:	c, d
c:	a, d
d:	d

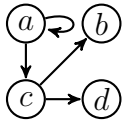
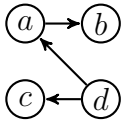
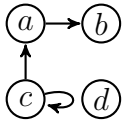
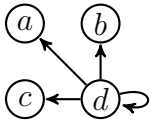
Sind $M_R = (r_{ij})$ und $M_S = (s_{ij})$ boolesche $n \times n$-Matrizen für R und S, so erhalten wir für $T = R \circ S$ die Matrix $M_T = (t_{ij})$ mit

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

Die Nachfolgermenge $T[x]$ 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 31. *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	$R \circ S$	$S \circ R$
Digraph				
Adjazenz-matrix	1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0	0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0	0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
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 $R \circ S = S \circ R$.

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 32. *Sei R eine Relation auf A.*

- *Die **reflexive Hülle** von R ist*

$$h_{refl}(R) = \bigcap\{S \subseteq A \times A \mid S \text{ ist reflexiv und } R \subseteq S\}.$$

- *Die **symmetrische Hülle** von R ist*

$$h_{sym}(R) = \bigcap\{S \subseteq A \times A \mid S \text{ ist symmetrisch und } R \subseteq S\}.$$

- *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*

$$R^* = \bigcap\{S \subseteq A \times A \mid S \text{ ist reflexiv, transitiv und } R \subseteq S\}.$$

- *Die **Äquivalenzhülle** von R ist*

$$h_{äq}(R) = \bigcap\{S \mid S \text{ ist eine Äquivalenzrelation auf } A \text{ und } R \subseteq S\}.$$

Satz 33. *Sei R eine Relation auf A.*

(i) $h_{refl}(R) = R \cup Id_A$,
(ii) $h_{sym}(R) = R \cup R^T$,
(iii) $R^+ = \bigcup_{n\geq 1} R^n$,
(iv) $R^* = \bigcup_{n\geq 0} R^n$,
(v) $h_{äq}(R) = (R \cup R^T)^*$.

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

$$x_0 R x_1 R x_2 \ldots x_{n-1} R x_n.$$

In der Graphentheorie nennt man $x_0, \ldots, x_n$ einen ***Weg*** der Länge n von a nach b. Ein Digraph G heißt ***zusammenhängend***, wenn es für je zwei Knoten a und b einen Weg von a nach b oder einen Weg von b nach a gibt. G heißt ***stark zusammenhängend***, wenn es von jedem Knoten a einen Weg zu jedem Knoten b in G gibt.

2.4.1 Ordnungs- und Äquivalenzrelationen

Wir betrachten zunächst Äquivalenzrelationen, die durch die drei Eigenschaften reflexiv, symmetrisch und transitiv definiert sind.

Ist E eine Äquivalenzrelation, so nennt man die Nachbarschaft $E[x]$ die ***von x repräsentierte Äquivalenzklasse*** und bezeichnet sie mit $[x]_E$ oder einfach mit $[x]$. Eine Menge $S \subseteq A$ heißt ***Repräsentantensystem***, falls sie genau ein Element aus jeder Äquivalenzklasse enthält.

Beispiel 34.

- *Auf der Menge aller Geraden im $\mathbb{R}^2$ die Parallelität. Offenbar bilden alle Geraden mit derselben Richtung (oder Steigung)*

jeweils eine Äquivalenzklasse. Daher wird ein Repräsentantensystem beispielsweise durch die Menge aller Ursprungsgeraden gebildet.

- *Auf der Menge aller Menschen "im gleichen Jahr geboren wie". Hier bildet jeder Jahrgang eine Äquivalenzklasse.*
- *Auf $\mathbb{Z}$ die Relation "gleicher Rest bei Division durch m". Die zugehörigen Äquivalenzklassen sind*

$$[r] = \{a \in \mathbb{Z} \mid a \equiv_m r\},\ r = 0, 1, \ldots, m-1.$$

Ein Repräsentantensystem wird beispielsweise durch die Reste $0, 1, \ldots, m-1$ gebildet. ◁

Die (bzgl. Inklusion) 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. $[x]_{Id_A} = \{x\}$ für alle $x \in A$, und die Allrelation erzeugt nur eine Äquivalenzklasse, nämlich $[x]_{A\times A} = A$ für jedes $x \in A$. Die Identität Id_A hat nur ein Repräsentantensystem, nämlich A. Dagegen kann jede Singletonmenge $\{x\}$ mit $x \in A$ als Repräsentantensystem für die Allrelation $A \times A$ fungieren.

Definition 35. *Eine Familie $\{B_i \mid i \in I\}$ von nichtleeren Teilmengen $B_i \subseteq A$ heißt* ***Partition*** *der Menge A, falls gilt:*

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

b) die Mengen B_i sind ***paarweise disjunkt****, d.h. für je zwei verschiedene Mengen $B_i \neq B_j$ gilt $B_i \cap B_j = \varnothing$.*

Wie der nächste Satz zeigt, bilden die Äquivalenzklassen einer Äquivalenzrelation E eine Partition $\{[x] \mid x \in A\}$ von A. Diese Partition wird auch ***Quotienten-*** oder ***Faktormenge*** genannt und mit A/E bezeichnet. Die Anzahl der Äquivalenzklassen von E wird auch als der ***Index*** von E bezeichnet.

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. E bewirkt also eine ***feinere*** Partitionierung als E'. Demnach ist die Identität die ***feinste*** und die Allrelation die ***gröbste*** Äquivalenzrelation.

Satz 36. *Sei E eine Relation auf A. Dann sind folgende Aussagen äquivalent.*

(i) E ist eine Äquivalenzrelation auf A.

(ii) Es gibt eine Partition $\{B_i \mid i \in I\}$ von A mit

$$xEy \Leftrightarrow \exists i \in I : x, y \in B_i.$$

Beweis.

$(i) \Rightarrow (ii)$ Sei E eine Äquivalenzrelation auf A. Wir zeigen, dass dann $\{E[x] \mid x \in A\}$ eine Partition von A mit der gewünschten Zusatzeigenschaft bildet:

Da E reflexiv ist, gilt xEx und somit $x \in E[x]$, d.h. $A = \bigcup_{x\in A} E[x]$.

Ist $E[x] \cap E[y] \neq \varnothing$ und $u \in E[x] \cap E[y]$, so folgt $E[x] = E[y]$:

$$z \in E[x] \Leftrightarrow xEz \overset{xEu}{\Leftrightarrow} uEz \overset{yEu}{\Leftrightarrow} yEz \Leftrightarrow z \in E[y]$$

Zudem gilt

$$\exists z \in A : x, y \in E[z] \Leftrightarrow \exists z : z \in E[x] \cap E[y]$$
$$\Leftrightarrow E[x] = E[y] \overset{y\in E[y]}{\Leftrightarrow} xEy$$

$(ii) \Rightarrow (i)$ Existiert umgekehrt eine Partition $\{B_i \mid i \in I\}$ von A mit $xEy \Leftrightarrow \exists i \in I : x, y \in B_i$, so ist E

- reflexiv, da zu jedem $x \in A$ eine Menge B_i mit $x \in B_i$ existiert,
- symmetrisch, da aus $x, y \in B_i$ auch $y, x \in B_i$ folgt, und
- transitiv, da aus $x, y \in B_i$ und $y, z \in B_j$ wegen $y \in B_i \cap B_j$ die Gleichheit $B_i = B_j$ und somit $x, z \in B_i$ folgt. ■

Als nächstes betrachten wir Ordnungsrelationen, die durch die drei Eigenschaften reflexiv, antisymmetrisch und transitiv definiert sind.

Beispiel 37.

- $(\mathsf{P}(M), \subseteq)$, $(\mathbb{Z}, \leq)$, $(\mathbb{R}, \leq)$ *und* $(\mathbb{N}, |)$ *sind Ordnungen.* $(\mathbb{Z}, |)$ *ist keine Ordnung, aber eine Quasiordnung.*
- *Für jede Menge* M *ist die relationale Struktur* $(\mathcal{P}(M); \subseteq)$ *eine Ordnung. Diese ist nur im Fall* $\|M\| \leq 1$ *linear.*
- *Ist* R *eine Relation auf* A *und* $B \subseteq A$*, so ist* $R_B = R \cap (B \times B)$ *die* Einschränkung *von* R *auf* B*.*
- *Einschränkungen von (linearen) Ordnungen sind ebenfalls (lineare) Ordnungen.*
- *Beispielsweise ist* $(\mathbb{Q}, \leq)$ *die Einschränkung von* $(\mathbb{R}, \leq)$ *auf* $\mathbb{Q}$ *und* $(\mathbb{N}, |)$ *die Einschränkung von* $(\mathbb{Z}, |)$ *auf* $\mathbb{N}$*.* ◁

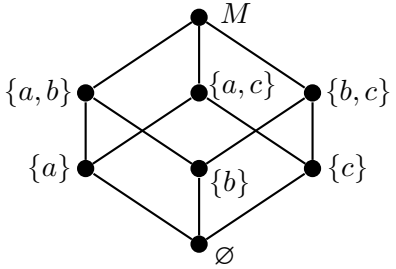
Ordnungen lassen sich sehr anschaulich durch Hasse-Diagramme darstellen. Sei $\leq$ eine Ordnung auf A und sei $<$ die Relation $\leq \cap \overline{Id}_A$. Um die Ordnung $\leq$ in einem ***Hasse-Diagramm*** darzustellen, wird nur der Graph der Relation

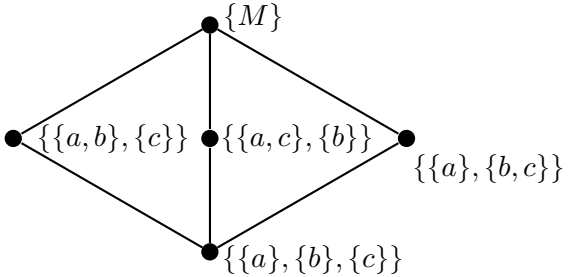
$$\lessdot = < \setminus <^2, \text{ d.h. } x \lessdot y \Leftrightarrow x < y \wedge \neg \exists z : x < z < y$$

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

Beispiel 38.

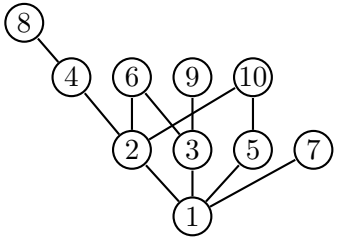
Das Hasse-Diagramm rechts zeigt die Inklusionsrelation auf der Potenzmenge $\mathcal{P}(M)$ *von* $M = \{a, b, c\}$*.*





Das Hasse-Diagramm links zeigt die feiner-Relation auf der Menge aller Partitionen von $M = \{a, b, c\}$*.*

Schränken wir die "teilt"-Relation auf die Menge $\{1, 2, \ldots, 10\}$ *ein, so erhalten wir nebenstehendes Hasse-Diagramm.*



◁

Definition 39. *Sei* $\leq$ *eine Ordnung auf* A *und sei* b *ein Element in einer Teilmenge* $B \subseteq A$*.*

- b *heißt* ***kleinstes Element*** *oder* ***Minimum*** *von* B *(kurz* $b = \min B$*), falls gilt:*

$$\forall b' \in B : b \leq b'.$$

- b *heißt* ***größtes Element*** *oder* ***Maximum*** *von* B *(kurz* $b = \max B$*), falls gilt:*

$$\forall b' \in B : b' \leq b.$$

- b *heißt* ***minimal*** *in* B*, falls es in* B *kein kleineres Element gibt:*

$$\forall b' \in B : b' \leq b \Rightarrow b' = b.$$

- b *heißt* ***maximal*** *in* B*, falls es in* B *kein größeres Element gibt:*

$$\forall b' \in B : b \leq b' \Rightarrow b = b'.$$

Bemerkung 40. *Da Ordnungen antisymmetrisch sind, kann es in jeder Teilmenge B höchstens ein kleinstes und höchstens ein größtes Element geben. Die Anzahl der minimalen und maximalen Elemente in B kann dagegen beliebig groß sein.*

Definition 41. *Sei $\leq$ eine Ordnung auf A und sei $B \subseteq A$.*

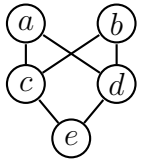
- *Jedes Element $u \in A$ mit $u \leq b$ für alle $b \in B$ heißt* **untere** *und jedes $o \in A$ mit $b \leq o$ für alle $b \in B$ heißt* **obere Schranke** *von B.*
- *B heißt* **nach oben beschränkt**, *wenn B eine obere Schranke hat, und* **nach unten beschränkt**, *wenn B eine untere Schranke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

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

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

$$(\forall b \in B : b \leq s) \wedge [\forall o \in A : (\forall b \in B : b \leq o) \Rightarrow s \leq o]$$

Beispiel 42. *Betrachte nebenstehende Ordnung. Die folgende Tabelle zeigt für verschiedene Teilmengen $B \subseteq \{a, b, c, d, e\}$ alle minimalen und maximalen Elemente, alle unteren und oberen Schranken sowie Minimum, Maximum, Infimum und Supremum von B (falls existent).*



B	minimal	maximal	min	max	untere Schranken	obere Schranken	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 43.

- *Es kann nicht mehr als ein Supremum und ein Infimum geben.*
- *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*

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

  weder ein Supremum noch ein Infimum.
- *Dagegen hat in $(\mathbb{R}, \leq)$ jede beschränkte Teilmenge ein Supremum und ein Infimum (aber eventuell kein Maximum oder Minimum).*

2.4.2 Abbildungen

Definition 44. *Sei R eine binäre Relation auf einer Menge M.*

- *R heißt* **rechtseindeutig**, *falls für alle $x, y, z \in M$ gilt:*

$$xRy \wedge xRz \Rightarrow y = z.$$

- *R heißt* **linkseindeutig**, *falls für alle $x, y, z \in M$ gilt:*

$$xRz \wedge yRz \Rightarrow x = y.$$

- *Der* **Nachbereich** $N(R)$ *und der* **Vorbereich** $V(R)$ *von* R *sind*
$$N(R) = \bigcup_{x \in M} R[x] \quad \text{und} \quad V(R) = \bigcup_{x \in M} R^T[x].$$
- *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 45.

- *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 46.

- *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****.*
- *Ist* f *injektiv, so ist auch* $f^{-1} : N(f) \to A$ *eine Abbildung, die als die zu* f inverse Abbildung *bezeichnet wird.*

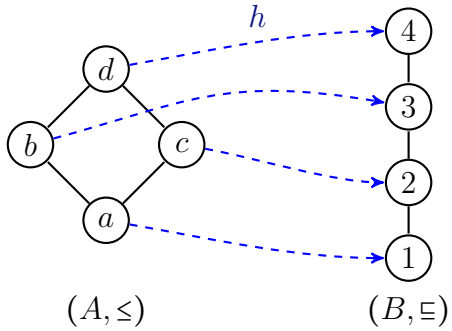
Man beachte, dass der Definitionsbereich $V(f^{-1}) = N(f)$ von f^{-1} nur dann gleich B ist, wenn f auch surjektiv, also eine Bijektion ist.

2.4.3 Homo- und Isomorphismen

Definition 47. *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 48. *Folgende Abbildung* $h : A_1 \to A_2$ *ist ein bijektiver Ordnungshomomorphismus.*

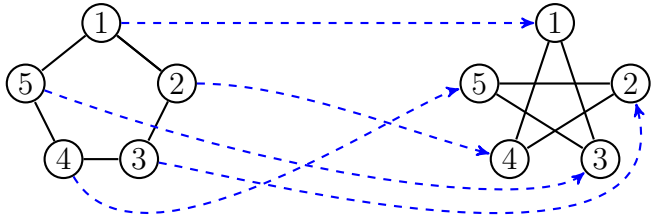


Obwohl h *ein bijektiver Homomorphismus ist, ist die Umkehrung* h^{-1} *kein Homomorphismus, da* h^{-1} *nicht monoton ist. Es gilt nämlich*
$$2 \sqsubseteq 3, \text{ aber } h^{-1}(2) = b \not\leq c = h^{-1}(3).$$
◁

Definition 49. *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 50.

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

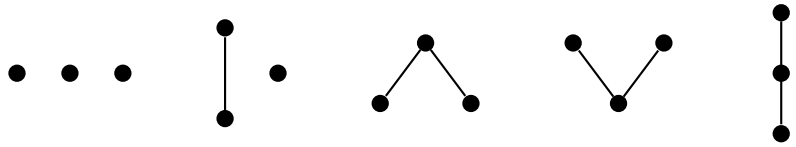


$G = (V, E)$ $\qquad$ $G' = (V, E')$

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

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

- *Die Abbildung $h : \mathbb{R} \to \mathbb{R}^+$ mit $h(x) = e^x$ ist ein Ordnungsisomorphismus zwischen $(\mathbb{R}, \leq)$ und $(\mathbb{R}^+, \leq)$.*
- *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.

- *Für $n \in \mathbb{N}$ sei $T_n = \{k \in \mathbb{N} \mid k \text{ teilt } n\}$ die Menge aller Teiler von n und $P_n = \{p \in T_n \mid p \text{ ist prim}\}$ die Menge aller Primteiler von n. Dann ist die Abbildung*

$$h : k \mapsto P_k$$

ein (surjektiver) Ordnungshomomorphismus von $(T_n, |)$ auf $(\mathcal{P}(P_n), \subseteq)$. h ist sogar ein Isomorphismus, falls n quadratfrei *ist (d.h. es gibt kein $k \geq 2$, so dass k^2 die Zahl n teilt).*

◁

2.5 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: $q \sim_M p$), wenn M von q und von p ausgehend jeweils dieselben Wörter akzeptiert. Bezeichnen wir den DFA $(Z, \Sigma, \delta, q, E)$ mit M_q, so sind q und p genau dann verschmelzbar, wenn $L(M_q) = L(M_p)$ ist. Offensichtlich ist $\sim_M$ eine Äquivalenzrelation.

Fassen wir alle mit einem Zustand z verschmelzbaren Zustände in dem neuen Zustand

$$[z]_{\sim_M} = \{z' \in Z \mid L(M_{z'}) = L(M_z)\}$$

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 $M' = (\tilde{Z}, \Sigma, \delta', \tilde{q}_0, \tilde{E})$ mit

$$\delta'(\tilde{q}, a) = \widetilde{\delta(q, a)}.$$

Hierbei bezeichnet $\tilde{Q}$ für eine Teilmenge $Q \subseteq Z$ die Menge $\{\tilde{q} \mid q \in Q\}$ aller Äquivalenzklassen $\tilde{q}$, die mindestens ein Element $q \in Q$ enthalten. Der nächste Satz zeigt, dass M' tatsächlich der gesuchte Minimalautomat ist.

Satz 51. *Sei $M = (Z, \Sigma, \delta, q_0, E)$ ein DFA, der nur Zustände enthält, die vom Startzustand q_0 aus erreichbar sind. Dann ist $M' = (\tilde{Z}, \Sigma, \delta', \tilde{q}_0, \tilde{E})$ mit*

$$\delta'(\tilde{q}, a) = \widetilde{\delta(q, a)}$$

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 $p \sim_M q$ auch $\delta(q, a)$ und $\delta(p, a)$ äquivalent sind:

$$\begin{aligned} L(M_q) = L(M_p) &\Rightarrow \forall x \in \Sigma^* : x \in L(M_q) \leftrightarrow x \in L(M_p) \\ &\Rightarrow \forall x \in \Sigma^* : ax \in L(M_q) \leftrightarrow ax \in L(M_p) \\ &\Rightarrow \forall x \in \Sigma^* : x \in L(M_{\delta(q,a)}) \leftrightarrow x \in L(M_{\delta(p,a)}) \\ &\Rightarrow L(M_{\delta(q,a)}) = L(M_{\delta(p,a)}). \end{aligned}$$

Als nächstes zeigen wir, dass $L(M') = L(M)$ ist. Sei $x = x_1 \dots x_n$ eine Eingabe und seien

$$q_i = \hat{\delta}(q_0, x_1 \dots x_i),\ i = 0, \dots, n$$

die von M bei Eingabe x durchlaufenen Zustände. Wegen

$$\delta'(\tilde{q}_{i-1}, x_i) = \widetilde{\delta(q_{i-1}, x_i)} = \tilde{q}_i$$

durchläuft M' dann die Zustände

$$\tilde{q}_0, \tilde{q}_1, \dots, \tilde{q}_n.$$

Da aber q_n genau dann zu E gehört, wenn $\tilde{q}_n \in \tilde{E}$ ist, folgt $L(M') = L(M)$ (man beachte, dass $\tilde{q}_n$ entweder nur Endzustände oder nur Nicht-Endzustände enthält, vgl. Beobachtung 52).

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 $k = \|\tilde{Z}\| = \|\{L(M_z) \mid z \in Z\}\|$ der Zustände von M' nicht von der Anzahl der Zustände von M, sondern nur von der erkannten Sprache $L = L(M)$ abhängt. Für $x \in \Sigma^*$ sei

$$L_x = \{y \in \Sigma^* \mid xy \in L\}$$

die ***Restsprache*** von L für das Wort x. Dann gilt $\{L_x \mid x \in \Sigma^*\} \subseteq \{L(M_z) \mid z \in Z\}$, da $L_x = L(M_{\hat{\delta}(q_0,x)})$ ist. Die umgekehrte Inklusion gilt ebenfalls, da nach Voraussetzung jeder Zustand $q \in Z$ über ein $x \in \Sigma^*$ erreichbar ist. Also hängt $k = \|\{L(M_z) \mid z \in Z\}\| = \|\{L_x \mid x \in \Sigma^*\}\|$ nur von L ab. ∎

Eine interessante Folgerung aus obigem Beweis ist, dass eine reguläre Sprache $L \subseteq \Sigma^*$ nur endlich viele verschiedene Restsprachen L_x, $x \in \Sigma^*$, hat. Daraus folgt, dass die durch

$$x \sim_L y \Leftrightarrow L_x = L_y$$

auf Σ^* definierte Äquivalenzrelation $\sim_L$ einen endlichen Index hat. Die Relation $\sim_L$ wird als *Nerode-Relation* von L bezeichnet.

Für die algorithmische Konstruktion von M' aus M ist es notwendig herauszufinden, ob zwei Zustände p und q von M äquivalent sind oder nicht. Hierzu genügt es, die Menge $D = \left\{\{p, q\} \subseteq Z \,\middle|\, p \not\sim_M q\right\}$ zu berechnen.

Bezeichne $A \triangle B = (A \setminus B) \cup (B \setminus A)$ die *symmetrische Differenz* von zwei Mengen A und B. Dann ist die Inäquivalenz $p \not\sim_M q$ zweier Zustände p und q gleichbedeutend mit $L(M_p) \triangle L(M_q) \neq \varnothing$. Wir nennen ein Wort $x \in L(M_p) \triangle L(M_q)$ einen *Unterscheider* zwischen p und q. Für $i \geq 0$ sei D_i die Menge aller Paare $\{p, q\}$, die einen Unterscheider x der Länge $|x| \leq i$ haben. Dann gilt $D = \bigcup_{i \geq 0} D_i$.

Beobachtung 52.

- *Das leere Wort ε unterscheidet Endzustände und Nichtendzustände, d.h.*

$$D_0 = \left\{\{p,q\} \subseteq Z \,\middle|\, p \in E, q \notin E\right\}.$$

- *Zudem unterscheidet ein Wort ax der Länge $i+1$ genau dann p und q, wenn x die beiden Zustände $\delta(p,a)$ und $\delta(q,a)$ unterscheidet, d.h.*

$$D_{i+1} = D_i \cup \left\{\{p,q\} \subseteq Z \,\middle|\, \exists a \in \Sigma : \{\delta(p,a), \delta(q,a)\} \in D_i\right\}.$$

Da es nur endlich viele Zustandspaare gibt, muss es ein k geben mit $D = D_k$. Offensichtlich gilt

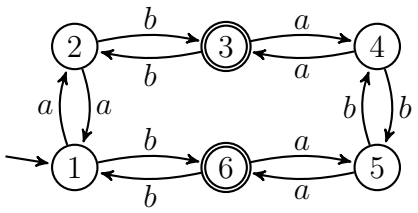
$$D = D_k \Leftrightarrow D_{k+1} = D_k.$$

Der folgende Algorithmus berechnet für einen beliebigen DFA M den zugehörigen Minimal-DFA M'.

Algorithmus `min-DFA`(M)

```
Input: DFA M = (Z, Σ, δ, q_0, E)
  entferne alle nicht erreichbaren Zustaende
  D' := {{z, z'} | z ∈ E, z' ∉ E}
  repeat
    D := D'
    D' := D ∪ {{p, q} | ∃a ∈ Σ : {δ(p, a), δ(q, a)} ∈ D}
  until D' = D
Output: M' = (Z̃, Σ, δ', q̃_0, Ẽ), wobei δ'(q̃, a) = δ̃(q, a) ist
 und für jeden Zustand q ∈ Z gilt: q̃ = {p ∈ Z | {p, q} ∉ D}
```

Beispiel 53. *Betrachte den DFA M:*



Dann enthält D_0 die Paare

$$\{1,3\},\{1,6\},\{2,3\},\{2,6\},\{3,4\},\{3,5\},\{4,6\},\{5,6\}.$$

Die Paare in D_0 sind in der folgenden Matrix durch den Unterscheider ε markiert.

2					
3	ε	ε			
4	a	a	ε		
5	a	a	ε		
6	ε	ε		ε	ε
	1	2	3	4	5

Wegen

$\{p,q\}$	$\{1,4\}$	$\{1,5\}$	$\{2,4\}$	$\{2,5\}$
$\{\delta(q,a),\delta(p,a)\}$	$\{2,3\}$	$\{2,6\}$	$\{1,3\}$	$\{1,6\}$

enthält D_1 zusätzlich die Paare $\{1,4\}$, $\{1,5\}$, $\{2,4\}$, $\{2,5\}$ (in obiger Matrix durch den Unterscheider a markiert). Da nun jedoch keines der verbliebenen Paare $\{1,2\}$, $\{3,6\}$, $\{4,5\}$ wegen

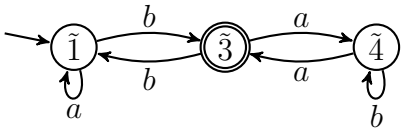
$\{p,q\}$	$\{1,2\}$	$\{3,6\}$	$\{4,5\}$
$\{\delta(p,a),\delta(q,a)\}$	$\{1,2\}$	$\{4,5\}$	$\{3,6\}$
$\{\delta(p,b),\delta(q,b)\}$	$\{3,6\}$	$\{1,2\}$	$\{4,5\}$

zu D_2 hinzugefügt werden kann, gilt $D_2 = D_1$ und somit $D = D_1$.

Aus den unmarkierten Paaren $\{1,2\}$, $\{3,6\}$ und $\{4,5\}$ erhalten wir die Äquivalenzklassen

$$\tilde{1} = \{1,2\},\ \tilde{3} = \{3,6\} \text{ und } \tilde{4} = \{4,5\},$$

die auf folgenden Minimal-DFA M' führen:



◁

Es ist auch möglich, einen Minimalautomaten M_L direkt aus einer regulären Sprache L zu gewinnen (also ohne einen DFA M für L zu kennen). Da wegen

$$\widetilde{\hat{\delta}(q_0,x)} = \widetilde{\hat{\delta}(q_0,y)} \Leftrightarrow \hat{\delta}(q_0,x) \sim_M \hat{\delta}(q_0,y)$$
$$\Leftrightarrow L(M_{\hat{\delta}(q_0,x)}) = L(M_{\hat{\delta}(q_0,y)}) \Leftrightarrow L_x = L_y$$

zwei Eingaben x und y den DFA M' genau dann in denselben Zustand überführen, wenn $L_x = L_y$ ist, können wir den von M' bei Eingabe x erreichten Zustand auch mit der Sprache L_x bezeichnen. Dies führt auf den zu M' isomorphen (also bis auf die Benennung der Zustände mit M' identischen) DFA $M_L = (Z_L, \Sigma, \delta_L, L_\varepsilon, E_L)$ mit

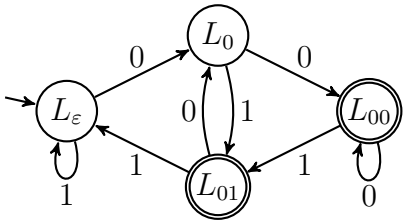
$$\begin{aligned} Z_L &= \{L_x \mid x \in \Sigma^*\}, \\ E_L &= \{L_x \mid x \in L\} \text{ und} \\ \delta_L(L_x, a) &= L_{xa}. \end{aligned}$$

M_L wird auch als ***Restsprachen-DFA*** für L bezeichnet.

Beispiel 54. *Die Sprache* $L = \{x_1 \dots x_n \in \{0,1\}^* \mid n \geq 2$ *und* $x_{n-1} = 0\}$ *hat die Restsprachen*

$$L_x = \begin{cases} L, & x \in \{\varepsilon, 1\} \text{ oder } x \text{ endet mit } 11, \\ L \cup \{0,1\}, & x = 0 \text{ oder } x \text{ endet mit } 10, \\ L \cup \{\varepsilon, 0, 1\}, & x \text{ endet mit } 00, \\ L \cup \{\varepsilon\}, & x \text{ endet mit } 01. \end{cases}$$

Somit erhalten wir den folgenden Minimalautomaten M_L *für* L*:*



Man beachte, dass es für die Konstruktion von M_L *keine Rolle spielt, wie die Restsprachen* L_x *konkret aussehen, d.h. ihre Angabe ist nicht erforderlich.* ◁

Notwendig und hinreichend für die Existenz von M_L ist, dass die Nerode-Relation $\sim_L$ von L endlichen Index hat bzw. L nur endlich viele verschiedene Restsprachen hat. Im Fall, dass M bereits ein Minimalautomat ist, sind alle Zustände von M' von der Form $\tilde{q} = \{q\}$, so dass M isomorph zu M' und damit auch isomorph zu M_L ist. Dies zeigt, dass alle Minimalautomaten für eine Sprache L isomorph sind.

Satz 55 (Myhill und Nerode)**.**

1. *Sei* L *regulär und sei* $\mathit{index}(\sim_L)$ *der Index von* $\sim_L$*. Dann gibt es für* L *bis auf Isomorphie genau einen Minimal-DFA. Dieser hat* $\mathit{index}(\sim_L)$ *Zustände.*
2. $\mathsf{REG} = \{L \mid$ *die Nerode-Relation* $\sim_L$ *hat endlichen Index*$\}$*.*

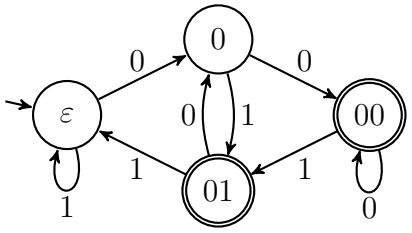
Sei R ein Repräsentantensystem für die Nerode-Relation $\sim_L$ von L, d.h. $\{L_x \mid x \in \Sigma^*\} = \{L_r \mid r \in R\}$ und $L_r \neq L_{r'}$ für alle $r, r' \in R$ mit $r \neq r'$. Dann können wir die Zustände des Minimal-DFA anstelle von L_x auch mit den Repräsentanten $r \in R$ bezeichnen. Dies führt auf den Minimal-DFA $M_R = (R, \Sigma, \delta, \varepsilon, E)$, wobei wir $\varepsilon \in R$ annehmen und $\delta(r,a) \in R$ der Repräsentant der Äquivalenzklasse $\widetilde{ra}$ und $E = R \cap L$ ist. Wir bezeichnen M_R als den zu R gehörigen ***Repräsentanten-DFA*** für L.

Beispiel 56. *Für die Sprache* $L = \{x_1 \dots x_n \in \{0,1\}^* \mid x_{n-1} = 0\}$ *lässt sich ein Repräsentanten-DFA* M_R *wie folgt konstruieren:*

1. *Wir beginnen mit* $r_1 = \varepsilon$.
2. *Da* $r_1 0 = 0 \not\sim_L \varepsilon$ *ist, erhalten wir* $r_2 = 0$ *und setzen* $\delta(\varepsilon, 0) = 0$.
3. *Da* $r_1 1 = 1 \sim_L \varepsilon$ *ist, setzen wir* $\delta(\varepsilon, 1) = \varepsilon$.
4. *Da* $r_2 0 = 00 \not\sim_L r_i$ *für* $i = 1, 2$ *ist, erhalten wir* $r_3 = 00$ *und setzen* $\delta(0, 0) = 00$.
5. *Da* $r_2 1 = 01 \not\sim_L r_i$ *für* $i = 1, 2, 3$ *ist, erhalten wir* $r_4 = 01$ *und setzen* $\delta(0, 1) = 01$.
6. *Da zudem* $r_3 0 = 000 \sim_L 00$, $r_3 1 = 001 \sim_L 01$, $r_4 0 = 010 \sim_L 0$ *und* $r_4 1 = 011 \sim_L \varepsilon$ *gilt, setzen wir* $\delta(00, 0) = 00$, $\delta(00, 1) = 01$, $\delta(01, 0) = 0$ *und* $\delta(01, 1) = \varepsilon$.

Wir erhalten also das Repräsentantensystem $R = \{\varepsilon, 0, 00, 01\}$ *für* $\sim_L$ *und folgenden Minimal-DFA* M_R *für* L*:*

r	ε	0	00	01
$\delta(r, 0)$	0	00	00	0
$\delta(r, 1)$	ε	01	01	ε



◁

Wir fassen nochmals die wichtigsten Ergebnisse zusammen.

Korollar 57. *Für jede Sprache* L *sind folgende Aussagen äquivalent:*

- L *ist regulär (d.h. es gibt einen DFA* M *mit* $L = L(M)$*),*
- *es gibt einen NFA* N *mit* $L = L(N)$*,*
- *es gibt einen regulären Ausdruck* γ *mit* $L = L(\gamma)$*,*
- L *hat endlich viele Restsprachen* $L_x = \{z \in \Sigma^* \mid xz \in L\}$, $x \in \Sigma^*$*,*
- *die Nerode-Relation* $\sim_L$ *von* L *hat endlichen Index.*

Wir können also beweisen, dass eine Sprache L **nicht** regulär ist, indem wir unendlich viele verschiedene Restsprachen (bzw. unendlich viele paarweise bzgl. $\sim_L$ inäquivalente Wörter) finden.

Satz 58. *Die Sprache* $L = \{a^n b^n \mid n \geq 0\}$ *ist nicht regulär.*

Beweis. Wegen

$$b^i \in L_{a^i} \triangle L_{a^j} \quad (\text{für } 0 \leq i < j)$$

sind die Restsprachen L_{a^i}, $i \geq 0$, paarweise verschieden und wegen

$$a^i \sim_L a^j \Leftrightarrow L_{a^i} = L_{a^j}$$

folgt auch, dass $a^i \not\sim_L a^j$ für $i < j$ gilt, weshalb $index(\sim_L) = \infty$ ist. ■

Wir werden im nächsten Abschnitt noch eine weitere Methode kennenlernen, mit der man beweisen kann, dass eine Sprache nicht regulär ist, nämlich das Pumping-Lemma.

2.6 Das Pumping-Lemma

Wie kann man von einer Sprache L noch nachweisen, dass sie nicht regulär ist? Eine weitere Möglichkeit besteht darin, die Kontraposition folgender Aussage anzuwenden.

Satz 59 (Pumping-Lemma für reguläre Sprachen)**.**
Zu jeder regulären Sprache L *gibt es eine Zahl* $l \geq 0$*, so dass sich alle Wörter* $x \in L$ *mit* $|x| \geq l$ *in* $x = uvw$ *zerlegen lassen mit*

1. $v \neq \varepsilon$*,*
2. $|uv| \leq l$ *und*
3. $uv^i w \in L$ *für alle* $i \geq 0$*.*

Falls eine Zahl $l \geq 0$ *mit diesen Eigenschaften existiert, wird das kleinste solche* l *die* ***Pumpingzahl*** *von* L *genannt.*

Beispiel 60. *Die Sprache*

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

hat die Pumpingzahl $l = 3$*. Sei nämlich* $x \in L$ *beliebig mit* $|x| \geq 3$*. Dann lässt sich innerhalb des Präfixes von* x *der Länge drei ein nichtleeres Teilwort* v *finden, das gepumpt werden kann:*

1. Fall: *x hat das Präfix ab (oder ba).*
Zerlege $x = uvw$ mit $u = \varepsilon$ und $v = ab$ (bzw. $v = ba$).

2. Fall: *x hat das Präfix aab (oder bba).*
Zerlege $x = uvw$ mit $u = a$ (bzw. $u = b$) und $v = ab$ (bzw. $v = ba$).

3. Fall: *x hat das Präfix aaa (oder bbb).*
Zerlege $x = uvw$ mit $u = \varepsilon$ und $v = aaa$ (bzw. $v = bbb$). ◁

Beispiel 61. *Eine endliche Sprache L hat die Pumpingzahl*

$$l = \begin{cases} 0, & L = \varnothing, \\ \max\{|x| + 1 \mid x \in L\}, & \text{sonst.} \end{cases}$$

Tatsächlich lässt sich jedes Wort $x \in L$ der Länge $|x| \geq l$ „pumpen“ (da solche Wörter gar nicht existieren), weshalb die Pumpingzahl höchstens l ist. Zudem gibt es im Fall $l > 0$ ein Wort $x \in L$ der Länge $|x| = l - 1$, das sich nicht „pumpen“ lässt, weshalb die Pumpingzahl nicht kleiner als l sein kann. ◁

Wollen wir mit Hilfe des Pumping-Lemmas von einer Sprache L zeigen, dass sie nicht regulär ist, so genügt es, für jede Zahl l ein Wort $x \in L$ der Länge $|x| \geq l$ anzugeben, so dass für jede Zerlegung von x in drei Teilwörter u, v, w mindestens eine der drei in Satz 59 aufgeführten Eigenschaften verletzt ist.

Beispiel 62. *Die Sprache*

$$L = \{a^j b^j \mid j \geq 0\}$$

ist nicht regulär, da sich für jede Zahl $l \geq 0$ das Wort $x = a^l b^l$ der Länge $|x| = 2l \geq l$ in der Sprache L befindet, welches offensichtlich nicht in Teilwörter u, v, w mit $v \neq \varepsilon$ und $uv^2w \in L$ zerlegbar ist. ◁

Beispiel 63. *Die Sprache*

$$L = \{a^{n^2} \mid n \geq 0\}$$

ist ebenfalls nicht regulär. Andernfalls müsste es nämlich eine Zahl l geben, so dass jede Quadratzahl $n^2 \geq l$ als Summe von natürlichen Zahlen $u + v + w$ darstellbar ist mit der Eigenschaft, dass $v \geq 1$ und $u + v \leq l$ ist, und für jedes $i \geq 0$ auch $u + iv + w$ eine Quadratzahl ist. Insbesondere müsste also $u + 2v + w = n^2 + v$ eine Quadratzahl sein, was wegen

$$n^2 < n^2 + v \leq n^2 + l < n^2 + 2l + 1 = (n + 1)^2$$

ausgeschlossen ist. ◁

Beispiel 64. *Auch die Sprache*

$$L = \{a^p \mid p \text{ prim }\}$$

ist nicht regulär, da sich sonst jede Primzahl p einer bestimmten Mindestgröße l als Summe von natürlichen Zahlen $u + v + w$ darstellen ließe, so dass $v \geq 1$ und für alle $i \geq 0$ auch $u + iv + w = p + (i - 1)v$ prim ist. Dies ist jedoch für $i = p + 1$ wegen

$$p + (p + 1 - 1)v = p(1 + v)$$

nicht der Fall. ◁

Bemerkung 65. *Mit Hilfe des Pumping-Lemmas kann nicht für jede Sprache $L \notin \mathsf{REG}$ gezeigt werden, dass L nicht regulär ist, da seine Umkehrung falsch ist. So hat beispielsweise die Sprache*

$$L = \{a^i b^j c^k \mid i = 0 \text{ oder } j = k\}$$

die Pumpingzahl 1 (d.h. jedes Wort $x \in L$ mit Ausnahme von ε kann „gepumpt“ werden). Dennoch ist L nicht regulär (siehe Übungen).

2.7 Grammatiken

Eine beliebte Methode, Sprachen zu beschreiben, sind Grammatiken. Implizit haben wir diese Methode bei der Definition der regulären Ausdrücke bereits benutzt.

Beispiel 66. *Die Sprache RA aller regulären Ausdrücke über einem Alphabet $\Sigma = \{a_1, \ldots, a_k\}$ lässt sich aus dem Symbol R durch wiederholte Anwendung folgender Ersetzungsregeln erzeugen:*

$$\begin{aligned} R &\to \varnothing, & R &\to RR, \\ R &\to \epsilon, & R &\to (R|R), \\ R &\to a_i,\ i = 1, \ldots, k, & R &\to (R)^*. \end{aligned}$$

◁

Definition 67. *Eine* ***Grammatik*** *ist ein 4-Tupel $G = (V, \Sigma, P, S)$, wobei*

- *V eine endliche Menge von* ***Variablen*** *(auch* ***Nichtterminalsymbole*** *genannt),*
- *Σ das* ***Terminalalphabet****,*
- *$P \subseteq (V \cup \Sigma)^+ \times (V \cup \Sigma)^*$ eine endliche Menge von* ***Regeln*** *(oder* ***Produktionen****) und*
- *$S \in V$ die* ***Startvariable*** *ist.*

Die Produktionenmenge P ist also eine binäre Relation auf $(V \cup \Sigma)^*$. Für $(u, v) \in P$ schreiben wir auch kurz $u \to_G v$ bzw. $u \to v$, wenn die benutzte Grammatik aus dem Kontext ersichtlich ist. Regeln der Form $\varepsilon \to v$ sind nicht erlaubt.

Definition 68. *Seien $\alpha, \beta \in (V \cup \Sigma)^*$.*

a) Wir sagen, ***β ist aus α in einem Schritt ableitbar*** *(kurz: $\alpha \Rightarrow_G \beta$), falls eine Regel $u \to_G v$ und Wörter $l, r \in (V \cup \Sigma)^*$ existieren mit*

$$\alpha = lur \text{ und } \beta = lvr.$$

Hierfür schreiben wir auch $l\underline{u}r \Rightarrow_G lvr$ bzw. $l\underline{u}r \Rightarrow lvr$.[†]

b) Die durch G ***erzeugte Sprache*** *ist*

$$L(G) = \{x \in \Sigma^* \mid S \Rightarrow^* x\}.$$

c) Ein Wort $\alpha \in (V \cup \Sigma)^*$ mit $S \Rightarrow^* \alpha$ heißt ***Satzform*** *von G.*

Beispiel 69. *Wir betrachten nochmals die Grammatik $G = (\{R\}, \Sigma \cup \{\varnothing, \epsilon, (,), ^*, |\}, P, R)$, die die Menge der regulären Ausdrücke über dem Alphabet Σ erzeugt, wobei P die oben angegebenen Regeln enthält. Ist $\Sigma = \{0, 1\}$, so lässt sich der reguläre Ausdruck $(01)^*(\epsilon|\varnothing)$ beispielsweise wie folgt in 8 Schritten ableiten:*

$$\begin{aligned} \underline{R} &\Rightarrow \underline{R}R \Rightarrow (\underline{R})^*R \Rightarrow (RR)^*R \Rightarrow (\underline{R}R)^*(R|R) \\ &\Rightarrow (0\underline{R})^*(R|R) \Rightarrow (01)^*(\underline{R}|R) \Rightarrow (01)^*(\epsilon|\underline{R}) \Rightarrow (01)^*(\epsilon|\varnothing) \end{aligned}$$

◁

Man unterscheidet vier verschiedene Typen von Grammatiken.

Definition 70. *Sei $G = (V, \Sigma, P, S)$ eine Grammatik.*

1. *G heißt vom* ***Typ 3*** *oder* ***regulär****, falls für alle Regeln $u \to v$ gilt: $u \in V$ und $v \in \Sigma V \cup \Sigma \cup \{\epsilon\}$.*
2. *G heißt vom* ***Typ 2*** *oder* ***kontextfrei****, falls für alle Regeln $u \to v$ gilt: $u \in V$.*
3. *G heißt vom* ***Typ 1*** *oder* ***kontextsensitiv****, falls für alle Regeln $u \to v$ gilt: $|v| \geq |u|$ (mit Ausnahme der ε-Sonderregel, siehe unten).*
4. *Jede Grammatik ist automatisch vom* ***Typ 0****.*

[†]Man beachte, dass durch Unterstreichen von u in α sowohl die benutzte Regel als auch die Stelle in α, an der u durch v ersetzt wird, eindeutig erkennbar sind. Da $\Rightarrow$ eine binäre Relation auf $(V \cup \Sigma)^*$ ist, bezeichnet $\Rightarrow^m$ das m-fache Relationenprodukt und $\Rightarrow^*$ die reflexive, transitive Hülle von $\Rightarrow$.

ε-Sonderregel: In einer kontextsensitiven Grammatik $G = (V, \Sigma, P, S)$ kann auch die verkürzende Regel $S \to \varepsilon$ benutzt werden. Aber nur, wenn das Startsymbol S nicht auf der rechten Seite einer Regel in P vorkommt.

Die Sprechweisen „vom Typ i“ bzw. „regulär“, „kontextfrei“ und „kontextsensitiv“ werden auch auf die durch solche Grammatiken erzeugte Sprachen angewandt. (Der folgende Satz rechtfertigt dies für die regulären Sprachen, die wir bereits mit Hilfe von DFAs definiert haben.) Die zugehörigen neuen Sprachklassen sind

$$\mathsf{CFL} = \{L(G) \mid G \text{ ist eine kontextfreie Grammatik}\},$$

(*context free languages*) und

$$\mathsf{CSL} = \{L(G) \mid G \text{ ist eine kontextsensitive Grammatik}\}$$

(*context sensitive languages*). Da die Klasse der Typ 0 Sprachen mit der Klasse der rekursiv aufzählbaren (*recursively enumerable*) Sprachen übereinstimmt, bezeichnen wir diese Sprachklasse mit

$$\mathsf{RE} = \{L(G) \mid G \text{ ist eine Grammatik}\}.$$

Die Sprachklassen

$$\mathsf{REG} \subset \mathsf{CFL} \subset \mathsf{CSL} \subset \mathsf{RE}$$

bilden eine Hierarchie (d.h. alle Inklusionen sind echt), die so genannte **Chomsky-Hierarchie**.

Als nächstes zeigen wir, dass sich mit regulären Grammatiken gerade die regulären Sprachen erzeugen lassen. Hierbei erweist sich folgende Beobachtung als nützlich.

Lemma 71. *Zu jeder regulären Grammatik $G = (V, \Sigma, P, S)$ gibt es eine äquivalente reguläre Grammatik G', die keine Produktionen der Form $A \to a$ hat.*

Beweis. Betrachte die Grammatik $G' = (V', \Sigma, P', S)$ mit

$$\begin{aligned} V' &= V \cup \{X_{neu}\}, \\ P' &= \{A \to aX_{neu} \mid A \to_G a\} \cup \{X_{neu} \to \varepsilon\} \cup P \setminus (V \times \Sigma). \end{aligned}$$

Es ist leicht zu sehen, dass G' die gleiche Sprache wie G erzeugt. ∎

Satz 72. $\mathsf{REG} = \{L(G) \mid G \text{ ist eine reguläre Grammatik}\}$.

Beweis. Sei $L \in \mathsf{REG}$ und sei $M = (Z, \Sigma, \delta, q_0, E)$ ein DFA mit $L(M) = L$. Wir konstruieren eine reguläre Grammatik $G = (V, \Sigma, P, S)$ mit $L(G) = L$. Setzen wir

$$\begin{aligned} V &= Z, \\ S &= q_0 \text{ und} \\ P &= \{q \to ap \mid \delta(q, a) = p\} \cup \{q \to \varepsilon \mid q \in E\}, \end{aligned}$$

so gilt für alle Wörter $x = x_1 \dots x_n \in \Sigma^*$:

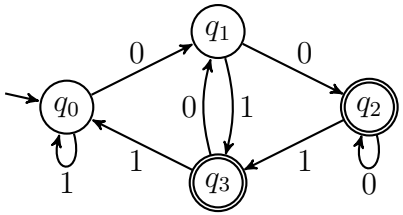
$$\begin{aligned} x \in L(M) &\Leftrightarrow \exists\, q_1, \dots, q_{n-1} \in Z\ \exists\, q_n \in E: \\ &\quad \delta(q_{i-1}, x_i) = q_i \text{ für } i = 1, \dots, n \\ &\Leftrightarrow \exists\, q_1, \dots, q_n \in V: \\ &\quad q_{i-1} \to_G x_i q_i \text{ für } i = 1, \dots, n \text{ und } q_n \to_G \varepsilon \\ &\Leftrightarrow \exists\, q_1, \dots, q_n \in V: \\ &\quad q_0 \Rightarrow_G^i x_1 \dots x_i q_i \text{ für } i = 1, \dots, n \text{ und } q_n \to_G \varepsilon \\ &\Leftrightarrow x \in L(G) \end{aligned}$$

Für die entgegengesetzte Inklusion sei nun $G = (V, \Sigma, P, S)$ eine reguläre Grammatik, die keine Produktionen der Form $A \to a$ enthält. Dann können wir die gerade beschriebene Konstruktion einer Grammatik aus einem DFA „umdrehen“, um ausgehend von G einen NFA $M = (Z, \Sigma, \delta, \{S\}, E)$ mit

$$\begin{aligned} Z &= V, \\ E &= \{A \mid A \to_G \varepsilon\} \text{ und} \\ \delta(A, a) &= \{B \mid A \to_G aB\} \end{aligned}$$

zu erhalten. Genau wie oben folgt nun $L(M) = L(G)$. ■ ◁

Beispiel 73. *Der DFA*



führt auf die Grammatik $(\{q_0, q_1, q_2, q_3\}, \{0, 1\}, P, q_0)$ *mit*

$$\begin{aligned} P: \quad & q_0 \to 1q_0, 0q_1, \\ & q_1 \to 0q_2, 1q_3, \\ & q_2 \to 0q_2, 1q_3, \varepsilon, \\ & q_3 \to 0q_1, 1q_0, \varepsilon. \end{aligned}$$

Umgekehrt führt die Grammatik $G = (\{A, B, C\}, \{a, b\}, P, A)$ *mit*

$$\begin{aligned} P: \quad & A \to aB, bC, \varepsilon, \\ & B \to aC, bA, b, \\ & C \to aA, bB, a \end{aligned}$$

über die Grammatik $G' = (\{A, B, C, D\}, \{a, b\}, P', A)$ *mit*

$$\begin{aligned} P': \quad & A \to aB, bC, \varepsilon, \\ & B \to aC, bA, bD, \\ & C \to aA, bB, aD, \\ & D \to \varepsilon \end{aligned}$$

auf den NFA